Exploring Expert Monosemanticity in MoE LLMs: Insights into Understanding and Fine-Tuning

Anonymous ACL submission

Abstract

Mixture-of-Experts (MoE) architectures enhance the scalability and efficiency of large language models (LLMs) by activating only a subset of parameters. However, the interpretability of individual experts and corresponding strategies for post-training adaptation to domain-specific tasks remain underexplored. In this work, we first develop an interpretability framework for expert-level monosemanticity in MoE models using sparse autoencoders, offering new insights into the specialization patterns of domain experts. Building on this, we propose an expert-frozen fine-tuning method that selectively updates domain-specific experts while keeping domain-agnostic experts fixed. We demonstrate that the inherent sparsity of MoE models encourages stronger monosemantic behavior at the expert level, which allows for the identification of experts responsible for particular downstream tasks and aids in preserving cross-domain performance. The proposed strategy further alleviates catastrophic forgetting and reduces computational overhead by limiting updates to domain-relevant experts. Experiments on specialized domains, including medical and legal corpora, show that our approach performs on par with or better than fully fine-tuned models on in-domain tasks, while achieving relative improvements of 21.19% and 60.58% in retaining performance on out-of-domain benchmarks. Compared to parameter-efficient fine-tuning baselines like LoRA, our method achieves superior performance on the target domain, while also achieving improvements of 11.29% and 52.70% in other domains.

1 Introduction

Mixture-of-Experts (MoE) architectures have recently emerged as a promising paradigm for scaling foundation models such as Kimi-K2, DeepSeek-V3 and Qwen3 (Team et al., 2025; Liu et al., 2024; Yang et al., 2025). By activating only a subset of experts in each layer, MoE substantially reduces computational overhead while preserving the overall capacity of the model. This sparsity allows training models with hundreds of billions of parameters without a proportional increase in inference cost, establishing MoE as a central design choice in state-of-the-art large-scale language models. Beyond efficiency, MoE offers a natural mechanism for expert specialization across different domains or tasks, which holds promise for improving the interpretability of these models (Jafar et al., 2025). Nevertheless, systematic explorations into how to better understand and leverage this interpretability remain limited.

A key property in analyzing model interpretability is monosemanticity, where each dimension is activated primarily by a single natural concept (Elhage et al., 2022; Cunningham et al., 2023), such as sentiment polarity, semantic categories, or word senses. The conventional approach to enhancing monosemanticity is to train a sparse autoencoder (SAE) on the internal activations of models. However, SAEs are post-hoc methods that introduce additional computational overhead. Prior work has shown that sparsity in the SAE feature space is closely linked to improved monosemanticity (Gao et al., 2024; Cunningham et al., 2023). This raises the question of whether MoE's inherent sparsity promotes monosemantic representations. In this paper, we propose a new analysis framework to evaluate the monosemanticity of MoE models in specific downstream domains. Our findings reveal that increased sparsity significantly strengthens expert-level monosemanticity and drives experts to specialize across different downstream domains.

Building on our analysis of enhanced monosemanticity in MoE models, we further examine how this property can inspire better post-training strategies for large language models. A well-known challenge in post-training, particularly supervised fine-tuning, is that optimizing for a specific downstream task often comes at the expense of performance

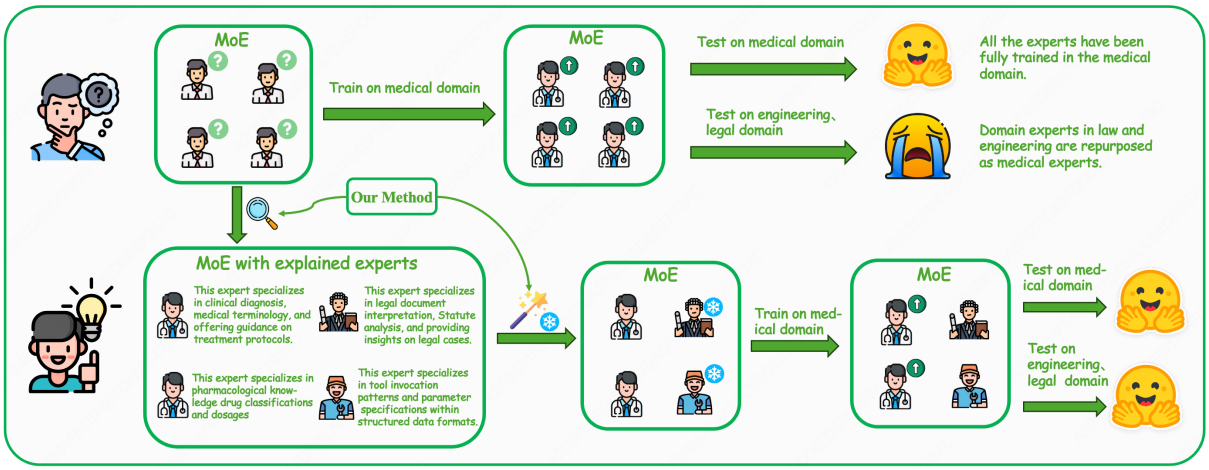


Figure 1: **(Top)** Training without our method: Expert functions remain uncharacterized, causing all experts to be fine-tuned on the target domain. This disrupts their original specializations, leading to degraded performance in other domains. **(Bottom)** Training with our method: We first generate explanations to reveal expert specializations, then leverage fine-grained metrics to selectively fine-tune relevant experts while freezing others. This preserves expert specialization and maintains performance across domains.

in other domains (Dong et al., 2023; Kumar et al., 2022). This raises the question of how to preserve cross-domain generalization while still achieving strong task-specific improvements. Monosemanticity provides a natural lens to address this issue: with MoE architectures, the increased sparsity encourages experts to specialize in distinct domains, suggesting that only a subset of experts is truly responsible for a given downstream task.

As shown in Figure 1, we obtained natural language explanations for expert functions and propose a fine-tuning strategy in which only the experts relevant to the target task and decoupled from other domains are fine-tuned, while the others remain frozen. Experimental results demonstrate that EDMRS achieves performance comparable to fully fine-tuned models on target tasks, while better preserving performance in other domains. By integrating monosemantic interpretability into post-training design, our approach offers a principled means of improving task-specific performance while mitigating negative transfer across domains. Our contributions can be summarized as follows:

- We introduce a novel framework to analyze monosemanticity in MoE LLMs, filling the gap in existing research on expert-level monosemanticity in MoE models and providing the infrastructure for further academic research in this area.
- Through detailed analysis, we enhance the understanding of MoE monosemanticity, showing that increased sparsity strengthens expert-level monosemanticity, with middle layers exhibiting the highest levels. Experts specialize differently across domains, and these specialization patterns are crucial for improving task performance and minimizing cross-domain interference.
- Leveraging these insights, we propose a fine-tuning strategy that combines expert monosemanticity, selectively optimizing experts relevant to specific downstream tasks, improving task performance while minimizing cross-domain interference.

2 Related Work & Preliminary

2.1 MoE LLMs

Sparse MoE LLMs have achieved significant scalability by activating only a subset of parameters in each layer through a routing network (Team et al., 2025; Liu et al., 2024). This modular design enables efficient scaling and introduces additional flexibility for model adaptation. Formally, in the l-th layer of the MoE, the output hidden state of the t-th token, h_t^l, is computed as:

$$h_t^l = \sum_{i=1}^{N} g_{i,t} \cdot \mathrm{FFN}_i(u_t^l) + u_t^l. \quad (1)$$

Here, N denotes the total number of experts, $\mathrm{FFN}_i(\cdot)$ represents the i-th expert network, and

$g_{i,t}$ is the gating value for the i-th expert when processing the t-th token. The gating mechanism is defined as:

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{TopK}(\{s_{j,t}\}_{j=1}^{N}, K) \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

where $s_{i,t}$ denotes the token-expert gating score that determines the routing of token t to expert i, and the function $\text{TopK}(\cdot, K)$ selects the K highest gating scores from the complete set of expert scores, effectively determining which experts will be activated for processing the current token.

2.2 Finetuning of MoE LLMs

Fine-tuning strategies for MoE LLMs are more complex than those for dense LLMs (Cai et al., 2025). Prior research has explored expert-specific tuning (Bai et al., 2025; Wang et al., 2024b), dynamic data mixing (Zhu et al., 2024), and router adaptation (Liu et al., 2025). A key challenge is identifying relevant experts for the target domain. Existing methods rely on black-box metrics—such as expert activation frequency or router output probabilities (Cai et al., 2025)—but these lack interpretability and fail to distinguish domain-specific from shared experts. As a result, the interpretability of individual experts in MoE LLMs remains underexplored. In contrast, we leverage interpretability principles from sparse autoencoders (Cunningham et al., 2023) to design interpretable expert-level fine-tuning strategies for MoE LLMs.

2.3 Monosemanticity in Large Language Models

LLMs perform well across domains (Liu et al., 2024; Yang et al., 2025), but their decision-making is opaque. Recent studies introduced SAEs (Gao et al., 2024; Cunningham et al., 2023), inducing monosemantic dimensions that capture specific concepts (Shu et al., 2025). SAEs have been effective in tasks like data selection (Ma et al., 2025), disentanglement (Pach et al., 2025), and privacy analysis (Frikha et al., 2025). They've also enabled interpretable reward models for transparent feedback (Zhang et al., 2025; Li et al., 2025). While focusing on dense LLMs, MoE models' expert-level structure remains underexplored. This paper explores expert-level monosemanticity in MoE LLMs and proposes interpretable fine-tuning strategies for better expert selection.

3 Expert-level Monosemanticity in MoE LLMs

SAEs have recently emerged as powerful tools for improving the interpretability of LLMs (Cunningham et al., 2023). SAEs encourage monosemanticity by enforcing sparsity, such that each hidden dimension tends to represent a distinct natural concept (Gao et al., 2024). Notably, MoE LLMs inherently exhibit sparsity through their expert selection mechanism, activating only a subset of experts for each input. This parallel motivates our central research question: Does the inherent sparsity in MoEs lead to stronger monosemanticity? To address this, Section 3.1 introduces a systematic framework for evaluating monosemanticity in MoE LLMs, while Section 3.2 provides empirical insights into how monosemanticity emerges within MoE architectures.

3.1 Analysis Framework for Monosemanticity in MoE LLMs

To investigate monosemanticity in MoE LLMs, we propose an analysis framework grounded in structural analogies between SAE and MoE architectures. Specifically, we map the sparse gating outputs in MoEs to the sparse intermediate representations in SAEs; the top-k activation mechanism in MoEs mirrors the thresholded activation in SAEs; and each MoE expert is considered analogous to a single SAE neuron. Building on these correspondences, we extend the notion of monosemanticity from the neuron level in SAEs to the expert level in MoE models. Inspired by automated SAE monosemanticity evaluation methods (Paulo et al., 2025), we present a systematic, expert-level analysis framework for MoE LLMs.

The proposed framework consists of two stages. First, we employ a three-stage sampling strategy to construct expert activation datasets, including top-activated samples, importance-weighted samples, and randomly selected negatives. Natural language explanations for each expert are then generated using two complementary prompts: P_{mono}, providing a general characterization, and $\text{P}^{(d)}_{domain}$, highlighting domain-specific semantics. In the second stage, an external LLM predicts expert activation based on these explanations, formulated as a binary classification task whose accuracy reflects the degree of expert-level monosemanticity. Complete prompt templates and design details are provided in Appendix A.1, with detailed expert explanations

in Appendix A.3.

To quantitatively assess the monosemanticity of MoE, we propose two key metrics: the expert monosemanticity score (EMS) and the expert domain monosemanticity score (EDMS). EMS measures the degree to which an expert exhibits monosemanticity across a general domain, while EDMS evaluates monosemanticity within a specific downstream domain. Formally, the EMS for expert E_i is defined as:

$$\text{EMS}_i = \frac{1}{|\mathcal{D}_i|} \sum_{s_j \in \mathcal{D}} \mathbb{I}[\hat{y}_{i,j}^{mono} = y_{i,j}], \quad (3)$$

where $\mathcal{D}$ denotes the test dataset and j indexes its samples. Here, $\hat{y}_{i,j}^{mono}$ represents the binary prediction of whether expert E_i would be activated for sample s_j, based on the explanation derived from prompt P_{mono}, and $y_{i,j}$ is the corresponding ground-truth activation label. Higher EMS values indicate stronger monosemanticity.

For evaluating monosemanticity in a specific domain d, the EDMS is defined as:

$$\text{EDMS}_i^{(d)} = \frac{1}{|\mathcal{D}_i|} \sum_{s_j \in \mathcal{D}} \mathbb{I}[\hat{y}_{i,j}^{(d)} = y_{i,j}^{(d)}]. \quad (4)$$

Here, $\hat{y}_{i,j}^{(d)}$ denotes the prediction of expert E_i for domain d, and $y_{i,j}^{(d)}$ is the corresponding ground-truth label. A high EDMS indicates that the expert is monosemantic with respect to specific concepts in the target domain.

3.2 Empirical Insights of Expert-level Monosemanticity

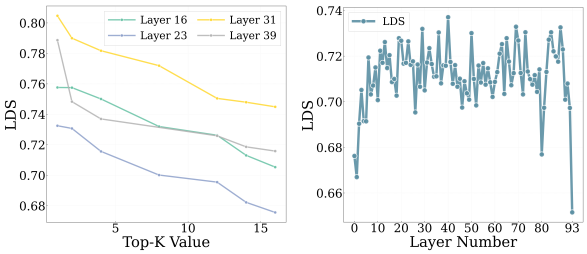
To advance the understanding of expert monosemanticity and sparsity in MoE models, and to lay the groundwork for further applications, we systematically analyze and uncover three key empirical insights in this section using the proposed analysis framework.

Insight 1: *The sparse architecture of MoE LLMs not only enhances computational efficiency but also induces a higher degree of expert monosemanticity, distinguishing them from dense LLMs.*

Clarification for Insight 1: Theoretical and empirical findings from SAE research establish that higher sparsity promotes stronger monosemanticity (Gao et al., 2024). In MoE LLMs, this sparsity is intrinsically determined by the number of activated experts, where a greater number of activations corresponds to lower sparsity. Building on this principle, our work investigates the relationship between expert-level sparsity and monosemanticity through controlled experiments on the Qwen3-30B-A3B-Instruct-2507 model (Yang et al., 2025). We select four representative layers and systematically vary the number of activated experts in each. Monosemanticity is quantified using the Layer-wise Decoupling Score (LDS), which we define as the average EMS of all experts within a layer:

$$\text{LDS}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \text{EMS}_i^{(l)}, \quad (5)$$

where LDS_l denotes the LDS for the l-th layer and N_l is the number of experts in that layer. LDS captures the average monosemanticity of experts at a given layer.



(a) Effect of Sparsity on LDS (b) LDS Values across Different Layers

Figure 2: Analysis of sparsity and monosemanticity in MoE. Layer-wise decoupling score (LDS) describes the average monosemanticity of a layer. Panel (a) examines the sparsity-monosemanticity relationship, showing that higher sparsity enhances expert specialization. Panel (b) tracks monosemanticity variation across network depth, revealing that middle layers exhibit optimal monosemanticity.

As shown in Figure 2a, increasing the number of activated experts (i.e., reducing sparsity) consistently and significantly decreases LDS across all layers, indicating progressively weaker functional separation and diminished expert-level monosemanticity. Building directly on this key observation, we next analyze how monosemanticity varies across different layers of the MoE architecture.

Insight 2: *Middle layers in MoE LLMs exhibit the highest degree of monosemanticity, while input and output layers show nearly the lowest. This suggests that experts in the middle layers are most decoupled and specialized, making them especially well-suited for operations that leverage monosemantic representations.*

Clarification for Insight 2: Figure 2b presents the LDS scores across network depth for Qwen3-235B-A22B-Instruct-2507. We observe a rapid

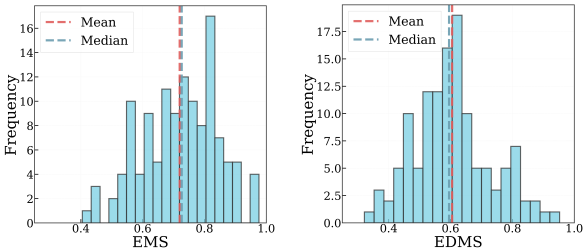
increase in LDS in the early layers, a consistently high plateau across the middle layers, and a sharp decline near the output. This trajectory suggests a three-stage organization: early layers focus on disentangling basic features, middle layers achieve the strongest expert decoupling and monosemanticity, and output layers gradually lose this separation. These findings highlight the central role of middle-layer experts in maintaining semantic specialization, i.e., monosemanticity, within the model. Besides patterns of monosemanticity across depth, we next examine domain-level monosemanticity to explore how expert specialization varies under different types of input data.To verify the stability of the metrics when using cheaper external LLMs, we added experiments in Appendix A.6.

Insight 3: *The degree of monosemanticity among MoE experts varies substantially across domains. Some experts exhibit consistently strong monosemanticity across all settings, while others are highly specialized within particular domains.*

Clarification for Insight 3: To investigate domain-dependent variations in expert monosemanticity, we apply our monosemanticity analysis framework using both P_{mono} and prompts $\mathrm{P}^{(d)}_{domain}$ to the middle layer of Qwen3-30B-A3B-Instruct-2507, computing EMS and EDMS for all 128 experts. As shown in Figure 3a and Figure 3b, EDMS scores are generally lower than EMS scores, suggesting that while some experts maintain high monosemanticity, their activity often pertains to domains outside of the evaluated domain. We also identify polysemantic experts who are activated across multiple domains. Overall, these findings reveal the coexistence of domain-specialized and shared experts, indicating that experts exhibit distinct patterns of activation across different domains.

4 Monosemanticity-Guided Fine-Tuning Strategy for MoE LLMs

During the fine-tuning process of MoE LLMs, it is widely observed that tuning on a target task will hurt the performance in other domains (Dong et al., 2023). Intuitively, as we have observed that there exist monosemantic experts, we can freeze the irrelevant experts and only update the experts related to the target domain to mitigate this phenomenon. In Section 4.1, we present our method for identifying monosemantic experts, and in Section 4.2, we provide a theoretical justification for this approach.



(a) Expert monosemanticity scores (b) Expert domain monosemanticity scores

Figure 3: Distribution of expert monosemanticity. Panel (a) evaluates the distribution of general expert monosemanticity scores (EMS) across all domains. Panel (b) evaluates the distribution of domain-specific expert domain monosemanticity scores (EDMS).

4.1 Expert Selection Strategy based on Monosemanticity Scores

In previous works (Bai et al., 2025; Wang et al., 2024b), the commonly used method is calculating the activation frequency (AF) to find important experts:

$$\mathrm{AF}_i^{(d)} = \frac{1}{|T_d|} \sum_{t \in T_d} \mathbb{I}[E_i'(t)], \tag{6}$$

where E_i' indicates whether expert E_i is activated for token t, T_d denotes the set of tokens in the target, and $\mathbb{I}[\cdot]$ is the indicator function. We know that there exist polysemantic experts who are activated across multiple semantics. Fine-tuning such experts will influence performance in other domains. Therefore, it is crucial to distinguish between monosemantic and polysemantic experts with high activation frequency.

Leveraging our analysis framework and the domain-specific prompt $\mathrm{P}^{(d)}_{domain}$, we can identify monosemantic experts that encode semantic features relevant to the target domain. These experts are key to domain adaptation and are less likely to interfere with other domains. Building on this, we introduce the expert domain monosemantic responsibility score (EDMRS) and its associated strategy. EDMRS combines an expert's activation frequency (AF) with its EDMS, weighting the importance of each expert's activations. Formally, the EDMRS for expert E_i in domain d is defined as:

$$\begin{aligned}\mathrm{EDMRS}_i^{(d)} &= \mathrm{AF}_i^{(d)} \times \mathrm{EDMS}_i^{(d)} \\ &= \frac{\mathrm{EDMS}_i^{(d)}}{|T_d|} \sum_{t \in T_d} \mathbb{I}[E_i'(t)]\end{aligned}$$

where $\text{EDMS}_i^{(d)}$ serves as a monosemanticity-based weighting coefficient, thereby increasing the importance of monosemanticity. A high EDMRS score indicates that the expert processes key semantic tokens relevant to the target and remains functionally decoupled from other domains.

Benefiting from the structural properties of MoE LLM and our design, our method can be orthogonally combined with fine-tuning approaches such as LoRA rather than being mutually exclusive. Since we essentially select parameters that are relevant to the target domain and decoupled from other domains, these fine-tuning methods can treat the selected parameters as a new sub-network; for example, LoRA's low-rank matrices can be directly applied to them (Hu et al., 2022).

In summary, the EDMRS strategy refines the traditional AF-based expert selection by explicitly incorporating both domain specificity and functional independence into the selection criterion, avoiding the disruption of general experts' distributions. By jointly considering an expert's activation frequency and its domain monosemanticity, EDMRS ensures that the selected experts are not only highly relevant to the target domain and make significant contributions but also minimally entangled with other domains. This targeted selection facilitates effective domain adaptation and fine-tuning, leading to enhanced performance in the target while substantially reducing cross-domain interference.

4.2 Theoretical Verification of EDMRS Strategy

As discussed earlier, we propose a new strategy to identify monosemantic experts related to a certain downstream domain. In this section, we provide a theoretical justification for our strategy.

Specifically, we denote d as the fine-tuned target domain, and consider a layer of the MoE model containing the expert set $E_1, E_2, \ldots, E_N$. For each expert E_i, we define its activation frequency in domain d as $\text{AF}_i^{(d)}$ and its Expert Domain Monosemanticity Score as $\text{EDMS}_i^{(d)}$. To formalize the objective for expert selection, we define the performance change resulting from fine-tuning expert E_i in domain d for a single input token t as:

$$C_i(t) = \Delta_{\mathcal{P}}^{(d)}(E_i, t) - \Delta_{\mathcal{N}}^{(\text{other})}(E_i, t) \quad (7)$$

where $\Delta_{\mathcal{P}}^{(d)}(E_i, t)$ represents the performance improvement of E_i on token t in the target d, while $\Delta_{\mathcal{N}}^{(\text{other})}(E_i, t)$ denotes the performance degradation of E_i on token t in other domains.

Within our MoE monosemanticity analysis framework, a high $\text{EDMS}_i^{(d)}$ score indicates that the token activation pattern of expert E_i aligns well with the domain-specific explanation. In other words, the token features processed by the expert are well-matched to target d, contributing substantially to domain-specific performance while remaining largely decoupled from other domains. Conversely, a low $\text{EDMS}_i^{(d)}$ implies that the expert's activations do not correspond to the domain explanation, with token features poorly aligned to d, resulting in minimal domain-specific contribution and increased coupling with other domains.

Building upon this theoretical intuition, we therefore posit the following hypothesis: that the average positive change in performance for a single token, when conditioned on the activation of expert E_i, exhibits a positive correlation with the value of $\text{EDMS}_i^{(d)}$.

Assumption 1.

$$\mathbb{E}[\Delta_{\mathcal{P}}^{(d)}(E_i, t) \mid E_i'] = k_1 \cdot \text{EDMS}_i^{(d)} \quad (8)$$

$$\mathbb{E}[\Delta_{\mathcal{N}}^{(other)}(E_i, t) \mid E_i'] = k_2 \cdot (1 - \text{EDMS}_i^{(d)}) \quad (9)$$

where $k_1, k_2 > 0$ are constants.

We next present the following theorem that characterizes the performance of our strategy:

Theorem 1 (EDMRS Optimal Expert Selection Theorem)**.** *Under Assumption 1, given a constraint on the number of selected experts m, selecting the m experts with the highest EDMRS scores $EDMRS_i^{(d)} = AF_i^{(d)} \cdot EDMS_i^{(d)}$ for fine-tuning maximizes the expected performance improvement in the target d:*

$$\begin{aligned} \mathrm{S}^* &= \arg\max_{|\mathrm{S}|=m} \sum_{i \in \mathrm{S}} \mathbb{E}[C_i] \\ &= \arg\max_{|\mathrm{S}|=m} \sum_{i \in \mathrm{S}} \text{EDMRS}_i^{(d)} \end{aligned}$$

where S^* denotes the optimal expert selection set, and $\mathbb{E}[C_i] = \mathbb{E}[C_i(t)]$ is the expected total performance change of expert E_i across all domains.

This theorem demonstrates that our EDMRS strategy theoretically maximizes performance both in the target and across other domains, providing a formal justification for our approach. A detailed proof is presented in Appendix A.4.

Method	biology	business	chemistry	computer science	economics	engineering	history	law	math	other	philosophy	physics	psychology	Average
Full	0.4686	0.3853	0.4364	0.2683	0.5284	0.3664	0.3911	0.2997	0.3494	0.3874	0.4148	0.4442	0.6529	0.4148
AF	0.6039	0.4284	0.4408	0.4951	0.5711	0.3736	0.4094	0.2852	0.4796	0.4773	0.3968	0.4742	0.5050	0.4570
EDMRS	0.6151	0.4715	0.4523	0.5537	0.6244	0.4118	0.5118	0.3143	0.5307	0.5011	0.4409	0.4788	0.6291	0.5027
vs Full	**+31.26%**	**+22.37%**	**+3.64%**	**+106.37%**	**+18.17%**	**+12.39%**	**+30.86%**	**+4.87%**	**+51.89%**	**+29.35%**	**+6.29%**	**+7.79%**	**-3.65%**	**+21.19%**
vs AF	**+1.85%**	**+10.06%**	**+2.61%**	**+11.84%**	**+9.33%**	**+10.22%**	**+25.01%**	**+10.20%**	**+10.65%**	**+4.99%**	**+11.11%**	**+0.97%**	**+24.57%**	**+10.00%**

(a) Performance comparison of models trained with EDMRS, AF and Full strategies on medical data in other domains.

Method	biology	business	chemistry	computer science	economics	engineering	health	history	math	other	philosophy	physics	psychology	Average
Full	0.0962	0.1014	0.1060	0.1000	0.1090	0.1053	0.1198	0.1181	0.0984	0.0963	0.1303	0.1109	0.1040	0.1074
AF	0.1255	0.1077	0.1051	0.0902	0.1137	0.5779	0.1308	0.1207	0.1132	0.1180	0.1423	0.1624	0.1253	0.1564
EDMRS	0.1409	0.1153	0.1201	0.0927	0.1268	0.5955	0.1333	0.2100	0.1244	0.1299	0.1403	0.1817	0.1303	0.1724
vs Full	**+46.47%**	**+13.71%**	**+13.30%**	**-7.30%**	**+16.33%**	**+465.53%**	**+11.27%**	**+77.82%**	**+26.42%**	**+34.89%**	**+7.67%**	**+63.84%**	**+25.29%**	**+60.58%**
vs AF	**+12.27%**	**+7.06%**	**+14.27%**	**+2.77%**	**+11.52%**	**+3.05%**	**+1.91%**	**+73.98%**	**+9.90%**	**+10.08%**	**-1.41%**	**+11.88%**	**+3.99%**	**+10.25%**

(b) Performance comparison of models trained with EDMRS, AF and Full strategies on legal data in other domains.

Method	biology	business	chemistry	computer science	economics	engineering	history	law	math	other	philosophy	physics	psychology	Average
LoRA	0.6109	0.4030	0.3993	0.4439	0.5983	0.3581	0.4514	0.2598	0.4123	0.4697	0.4369	0.4357	0.5927	0.4517
EDMRS	0.6151	0.4715	0.4523	0.5537	0.6244	0.4118	0.5118	0.3143	0.5307	0.5011	0.4409	0.4788	0.6291	0.5027
vs LoRA	**+0.69%**	**+17.00%**	**+13.27%**	**+24.74%**	**+4.36%**	**+15.00%**	**+13.38%**	**+20.98%**	**+28.72%**	**+6.69%**	**+0.92%**	**+9.89%**	**+6.14%**	**+11.29%**

(c) Performance comparison of PEFT-trained models on medical data across other domains.

Method	biology	business	chemistry	computer science	economics	engineering	health	history	math	other	philosophy	physics	psychology	Average
LoRA	0.1423	0.1039	0.0963	0.1268	0.1161	0.1166	0.1015	0.1181	0.1029	0.1158	0.1242	0.1024	0.1003	0.1129
EDMRS	0.1409	0.1153	0.1201	0.0927	0.1268	0.5955	0.1333	0.2100	0.1244	0.1299	0.1403	0.1817	0.1303	0.1724
vs LoRA	**-0.98%**	**+10.97%**	**+24.71%**	**-26.89%**	**+9.22%**	**+410.72%**	**+31.33%**	**+77.82%**	**+20.89%**	**+12.18%**	**+12.96%**	**+77.44%**	**+29.91%**	**+52.70%**

(d) Performance comparison of PEFT-trained models on legal data across other domains.

Table 1: Comprehensive performance comparison across different domains and training strategies: (a) Trained on medical domain data, comparing AF and Full strategies; (b) Trained on legal domain data, comparing AF and Full strategies; (c) Trained on medical domain data, comparing PEFT strategy; (d) Trained on legal domain data, comparing PEFT strategy.

5 Experiments

EDMRS-selected experts	AF-selected experts
Expert 111: Medical terminology and pathology-specific vocabulary including disease mechanisms and clinical references	Expert 116: This expert specializes in processing exact word patterns, numerical sequences, and precise linguistic structures within technical or instructional contexts
Expert 53: This expert specializes in content related to human anatomy and physical movements emphasizing injury mechanisms and clinical examination techniques	Expert 72: The expert specializes in technical and scientific content involving mathematical calculations and physical properties

Table 2: Examples of differences between experts selected by the EDMRS strategy and the AF strategy for the medical domain, where EDMRS favors monosemantic experts while AF selects general experts.

This section validates the effectiveness of our proposed EDMRS strategy through two sets of experiments. The first set aims to answer: compared to the most commonly used full fine-tuning method (Full Strategy) and the expert selection method based on activation frequency (AF Strategy), does EDMRS offer greater advantages? The second set of experiments investigates: compared to commonly used parameter-efficient fine-tuning (PEFT) methods, how competitive is our EDMRS strategy as an independent fine-tuning approach?

We conduct experiments using MedMCQA and LegalBench datasets, alongside out-of-domain subsets from MMLU-Pro, including **13 specialized domains** like economics, math, and history (Pal et al., 2022; Guha et al., 2023; Wang et al., 2024a). Further experimental details are in Appendix A.5.

5.1 Comparison with Full and AF Strategies

To evaluate the impact of post-training on performance in other domains, we perform expert-selection fine-tuning for the medical domain and legal domain. We compare the effectiveness of our proposed EDMRS strategy against conventional approaches by considering three strategies: (1) Full-parameter Strategy: all model parameters are fine-

tuned, serving as the baseline; (2) AF Strategy: fine-tune the k experts with the highest activation frequencies; (3) EDMRS Strategy: fine-tune the k experts with the highest EDMRS.

Method	Average Accuracy on MedMCQA	Average Accuracy on MMLU-Pro
Full	60.72%	41.48%
AF	48.48%	45.70%
EDMRS	60.34%	50.27%
vs Full	**-0.63%**	**+21.19%**
vs AF	**+24.46%**	**+10.00%**

Table 3: Comparative analysis of models fine-tuned on medical data.

Method	Average Accuracy on LegalBench	Average Accuracy on MMLU-Pro
Full	72.38%	10.74%
AF	76.58%	15.64%
EDMRS	76.79%	17.24%
vs Full	**+6.09%**	**+60.58%**
vs AF	**+0.27%**	**+10.25%**

Table 4: Comparative analysis of models fine-tuned on legal data.

See results in Table 3, EDMRS (60.34%) outperforms AF (48.48%) and approaches Full (60.72%). More importantly, EDMRS demonstrates better performance in other domains (50.27%), surpassing the Full (41.48%) and AF (45.70%) strategies by relative margins of 21.19% and 10.00%, respectively. Table 1(a) shows EDMRS is clearly the best. Notably, EDMRS outperforms the Full by 51.89% on math, reflecting that EDMRS preserves important reasoning capabilities. Table 4 and Table1(b) shows consistent advantages are also observed in the legal domain, surpassing the Full and AF strategies by relative margins of 60.58% and 10.25%.

Table 2 further reveals that EDMRS selects experts specialized for the target domain, unlike AF, which tends to select general experts. This advantage of EDMRS is due to its ability to avoid the impact of general experts by only updating domain-specific ones, reducing the loss of capabilities in other domains.

5.2 Comparison with PEFT as a Standalone Strategy

Method	Average Accuracy on MedMCQA	Average Accuracy on MMLU-Pro
LoRA	0.5090	0.4517
EDMRS	0.6034	0.5027
vs LoRA	**+18.55%**	**+11.29%**

Table 5: Comparative analysis of EDMRS and LoRA for model fine-tuning on medical data.

Method	Average Accuracy on LegalBench	Average Accuracy on MMLU-Pro
LoRA	0.6559	0.1129
EDMRS	0.7679	0.1724
vs LoRA	**+17.08%**	**+52.70%**

Table 6: Comparative analysis of EDMRS and LoRA for model fine-tuning on legal data.

It should be noted that our method is orthogonal to PEFT approaches, such as applying LoRA's low-rank matrices directly to the experts we select. To compare their standalone capabilities, we evaluate against LoRA. As shown in Tables 5, 6, 1(c) and 1(d), EDMRS achieves better performance on the target domain (by 18.55% for medical and 17.08% for legal) while outperforming LoRA across almost all other domains and on average (by 11.29% for medical and 52.70% for legal), further demonstrating the advantages of EDMRS. This is because LoRA still updates all experts, whereas our strategy more effectively prevents cross-domain interference by freezing irrelevant experts.

5.3 Overall Discussion

Taking into account both the comparable performance in the target and the improvements on retention across other domains, the experimental results demonstrate the substantial advantage of EDMRS, which mitigates cross-domain interference caused by parameter updates at the semantic level, ultimately achieving enhanced target-domain performance while alleviating catastrophic forgetting.

6 Conclusion

We establish an analysis framework that extends SAE monosemanticity analysis to MoE and propose a fine-tuning strategy guided by monosemanticity analysis, emphasizing the role of sparsity in fostering expert specialization across domains. We define four novel metrics to assess monosemanticity and domain alignment, and use these to develop a selective fine-tuning method that updates only relevant experts, while freezing unrelated ones. Experiments show that our method significantly enhances performance retention and generalization across other domains. We believe that our framework provides a principled approach for understanding and leveraging the monosemanticity of MoE, with potential applicability to a wide range of real-world scenarios.

7 Limitations

Our method introduces some performance overhead due to the process of generating expert-level explanations and calculating monosemanticity scores. Additionally, we have not explored the orthogonal combination with other parameter-efficient fine-tuning (PEFT) methods, which could potentially enhance performance.

References

Jun Bai, Minghao Tong, Yang Liu, Zixia Jia, and Zilong Zheng. 2025. Understanding and leveraging the expert specialization of context faithfulness in mixture-of-experts llms. *arXiv preprint arXiv:2508.19594*.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. 2025. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Ahmed Frikha, Muhammad Reza Ar Razi, Krishna Kanth Nakka, Ricardo Mendes, Xue Jiang, and Xuebing Zhou. 2025. Privacyscalpel: Enhancing llm privacy via interpretable feature intervention with sparse autoencoders. *arXiv preprint arXiv:2503.11232*.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Rajab Jafar, Fawzi Gamal, and Rais Raheem. 2025. Scalable and interpretable mixture of experts models in machine learning: Foundations, applications, and challenges.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*.

Sihang Li, Wei Shi, Ziyuan Xie, Tao Liang, Guojun Ma, and Xiang Wang. 2025. Safer: Probing safety in reward models with sparse autoencoder. *arXiv preprint arXiv:2507.00665*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Yilun Liu, Yunpu Ma, Yuetian Lu, Shuo Chen, Zifeng Ding, and Volker Tresp. 2025. Parameter-efficient routed fine-tuning: Mixture-of-experts demands mixture of adaptation modules. *arXiv preprint arXiv:2508.02587*.

Da Ma, Gonghu Shang, Zhi Chen, Libo Qin, Yijie Luo, Lei Pan, Shuai Fan, Lu Chen, and Kai Yu. 2025. Task-specific data selection for instruction tuning via monosemantic neuronal activations. *arXiv preprint arXiv:2503.15573*.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. 2025. Sparse autoencoders learn monosemantic features in vision-language models. *arXiv preprint arXiv:2504.02821*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. 2025. Automatically interpreting millions of features in large language models. *Preprint*, arXiv:2410.13928.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. 2025. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. *arXiv preprint arXiv:2503.05613*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max W.F. Ku, Kai Wang, Alex Zhuang, Rongqi "Richard" Fan, Xiang Yue, and Wenhu Chen. 2024a. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *ArXiv*, abs/2406.01574.

Zihan Wang, Deli Chen, Damai Dai, Runxin Xu, Zhuoshu Li, and Yu Wu. 2024b. Let the expert stick to his last: Expert-specialized fine-tuning for sparse architectural large language models. *arXiv preprint arXiv:2407.01906*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shuyi Zhang, Wei Shi, Sihang Li, Jiayi Liao, Tao Liang, Hengxing Cai, and Xiang Wang. 2025. Interpretable reward model via sparse autoencoder. *arXiv preprint arXiv:2508.08746*.

Tong Zhu, Daize Dong, Xiaoye Qu, Jiacheng Ruan, Wenliang Chen, and Yu Cheng. 2024. Dynamic data mixing maximizes instruction tuning for mixture-of-experts. *arXiv preprint arXiv:2406.11256*.

A Appendix

A.1 Details of Prompts Used in Our monosemanticity analysis framework

This section provides a detailed introduction to the prompt designs used in our expert monosemanticity analysis framework, covering both explanation generation and evaluation stages. The prompt contents are crucial for the accuracy of expert explanation generation and evaluation, and facilitate reproducibility and comparison.

A.1.1 Explanation Generation Prompts: P_{mono} and $\mathrm{P}^{(d)}_{domain}$

During the explanation generation stage, we design two types of prompts: P_{mono} and $\mathrm{P}^{(d)}_{domain}$. Full details are as follows:

Full Details of P_{mono}

System:
We're studying experts in a large-scale Mixture of Experts (MoE) language model. Each expert specializes in processing certain types of content, knowledge domains, linguistic patterns, or reasoning tasks in documents. The content that activates each expert in documents is indicated with << ... >>. We will give you a list of documents on which the expert is activated by the router, in order from most strongly activating to least strongly activating. Look at the parts of the document the expert activates for and summarize in a single sentence what type of content or domain this expert specializes in. Provide a reasonable explanation that captures the key distinguishing features. Note that some experts will specialize in very specific linguistic patterns or substrings, but others will activate on broader content types provided that the text contains particular knowledge domains or reasoning patterns. Your explanation should cover the general specialization pattern across most or all activating content (for example, don't give an explanation specific to a single domain if the expert activates across multiple related areas). Pay attention to things like the subject matter, linguistic style, technical complexity, or reasoning type of the activating content, if that seems relevant. Keep the explanation as short and simple as possible, limited to 20 words or fewer. Omit punctuation and formatting. You should avoid giving long lists of specific topics. Include specific details about the semantic patterns, syntactic structures, or functional roles that characterize this expert's specialization. Mention key distinguishing features such as specific word types, grammatical constructions, or conceptual themes.

Your response should be in the form "This expert specializes in...".

User:
Here is the explanation: this expert specializes in

Here are the examples:
{examples}

Here, {examples} refers to our sampled activation examples, with activated tokens marked using <<>>.

Full Details of $\mathrm{P}_{domain}^{(d)}$

$\mathrm{P}_{domain}^{(d)}$ includes all the content of P_{mono}. Additionally, to constrain $\mathrm{P}_{domain}^{(d)}$ to generate explanations within a specific domain, taking the medical and legal domain as an example, we add the following prompt on the basis of P_{mono}:

System:
For the medical dataset:
This expert processes medical multiple-choice questions and clinical reasoning, so focus on medical terminology, diagnostic reasoning patterns, clinical decision-making, anatomical concepts, pharmacological knowledge, pathophysiology, treatment protocols, medical examination patterns, or healthcare domain knowledge.

IMPORTANT: This expert should ONLY specialize in medical and healthcare content. Do not generate explanations about non-medical domains.

For the legal dataset:
This expert processes legal content and reasoning, so focus on legal terminology, case law analysis, statutory interpretation, legal argumentation patterns, contractual language, judicial reasoning, legal precedents, or jurisprudential concepts.

IMPORTANT: This expert should ONLY specialize in legal and jurisprudence content. Do not generate explanations about non-legal domains.

Your response should be in the form "This expert specializes in...".

For the medical dataset:

Some examples: "This expert specializes in differential diagnosis reasoning in multiple-choice contexts", "This expert specializes in pharmacological drug mechanism descriptions and interactions", "This expert specializes in anatomical structure identification and spatial relationships", "This expert specializes in clinical symptom pattern recognition and analysis", "This expert specializes in medical procedure descriptions and protocols", "This expert specializes in pathophysiology and disease mechanism explanations", "This expert specializes in medical terminology and nomenclature", "This expert specializes in treatment option evaluation and selection".

For the legal dataset:

Some examples: "This expert specializes in contract clause interpretation and analysis", "This expert specializes in case law citation and precedent application", "This expert specializes in statutory construction and legislative intent", "This expert specializes in legal doctrine and principle formulation", "This expert specializes in evidentiary reasoning and burden of proof", "This expert specializes in constitutional law interpretation", "This expert specializes in tort law causation analysis", "This expert specializes in criminal law mens rea and actus reus concepts".

A.2 Explanation Evaluation Prompts

In the explanation evaluation stage, the full details of the designed prompts are as follows:

Full Details of Explanation Evaluation Prompts

System:
We're studying experts in a large-scale Mixture of Experts (MoE) language model. Each expert in the MoE layers specializes in processing certain types of content, knowledge domains, linguistic patterns, or reasoning tasks.

You will be given a short explanation of what this expert specializes in, and then be shown N example sequences in random order.

You will have to return a comma-separated list of the examples where you think this expert should be activated by the router, based on the content type, domain knowledge, or linguistic characteristics required.

For example, your response might look like "1, 3, 5".

Try not to be overly specific in your interpretation of the explanation.

If you think there are no examples where this expert will be significantly activated, you should just respond with "None".

You should include nothing else in your response other than comma-separated numbers or the word "None" - this is important.
User:

The documents which this expert is activated are given below:

{examples}

Here, {examples} is a set of numbered samples, where each sample is shown in its original form without <<>> indicating activated tokens.

General Expert Explanations by P_{mono}

- This expert specializes in processing geographical coordinates and East Asian country names and language codes.
- This expert specializes in processing scientific and mathematical content such as synaptic input, quantum physics, and neuronal activity rate.
- This expert specializes in content related to gas and electricity prices and environmental impact on air quality.
- This expert specializes in documentation or metadata with type declarations and default value specifications.
- This expert specializes in processing scientific measurements and data conversion parameters in technical domains.
- The expert specializes in technical and scientific content involving mathematical calculations and physical properties.
- This expert specializes in processing exact word patterns, numerical sequences, and precise linguistic structures within technical or instructional contexts.

A.3 Expert Explanations from the Monosemanticity Analysis Framework

To enhance the understanding of expert specialization within MoE, we present the natural language explanations generated by our monosemanticity analysis framework below.

Medical Domain Expert Explanations by $\mathrm{P}^{(d)}_{domain}$

- This expert specializes in preventive and social medicine textbook content analysis
- This expert specializes in diagnostic and treatment guidance for obstetrics and pediatrics in medical contexts
- This expert specializes in clinical reasoning for medical multiple-choice questions focusing on diagnostic and treatment options
- This expert specializes in medical legal and forensic terminology and concepts in healthcare contexts
- This expert specializes in step-by-step diagnostic reasoning in medical multiple-choice question contexts

Legal Domain Expert Explanations by $\mathrm{P}^{(d)}_{domain}$

- This expert specializes in corporate law structures and governance, including company incorporation, executive roles, and shareholder rights
- This expert specializes in legal clause interpretation and statutory meaning analysis
- This expert specializes in compassionate use programs and legal aspects of drug administration outside clinical trials
- This expert specializes in contractual clauses and legal agreement interpretation
- This expert specializes in privacy policy language and data collection clauses in contractual agreements

A.4 Detailed Proof of Theorem 1

The detailed proof of Theorem 1 is provided below.

Proof. By the law of total expectation, the expected total performance change for expert E_i across all domains is given by:

$$\begin{aligned}\mathbb{E}[C_i] &= \mathbb{E}[C_i(t)] \\ &= \mathbb{E}[\Delta_{\mathcal{P}}^{(d)}(E_i,t)] - \mathbb{E}[\Delta_{\mathcal{N}}^{(\text{other})}(E_i,t)].\end{aligned}$$

Within the sparse activation mechanism of the MoE architecture, an expert only contributes to the model output when it is activated. If the expert is not activated, it does not induce any performance change, i.e.,

$$\mathbb{E}[\Delta_{\mathcal{P}}^{(d)}(E_i,t) \mid E_i''] = 0 \tag{10}$$

$$\mathbb{E}[\Delta_{\mathcal{N}}^{(\text{other})}(E_i,t) \mid E_i''] = 0 \tag{11}$$

where E_i'' indicates that expert E_i is not activated during the output of token t.

Therefore, by the law of total expectation, the expectation can be decomposed into the cases of activation and non-activation:

$$\begin{aligned}\mathbb{E}[\Delta_{\mathcal{P}}^{(d)}(E_i,t)] &= P(E_i') \cdot \mathbb{E}[\Delta_{\mathcal{P}}^{(d)}(E_i,t) \mid E_i'] \\ &\quad + P(E_i'') \cdot \mathbb{E}[\Delta_{\mathcal{P}}^{(d)}(E_i,t) \mid E_i''] \\ &= \mathrm{AF}_i^{(d)} \cdot \mathbb{E}[\Delta_{\mathcal{P}}^{(d)}(E_i,t) \mid E_i'] \\ &\quad + (1 - \mathrm{AF}_i^{(d)}) \cdot 0 \\ &= \mathrm{AF}_i^{(d)} \cdot \mathbb{E}[\Delta_{\mathcal{P}}^{(d)}(E_i,t) \mid E_i']\end{aligned} \tag{12}$$

Similarly,

$$\mathbb{E}[\Delta_{\mathcal{N}}^{(\text{other})}(E_i,t)] = \mathrm{AF}_i^{(d)} \cdot \mathbb{E}[\Delta_{\mathcal{N}}^{(\text{other})}(E_i,t) \mid E_i'] \tag{13}$$

Thus, the expected total performance change is given by:

$$\begin{aligned}\mathbb{E}[C_i] = \mathrm{AF}_i^{(d)} \cdot \Big[&\mathbb{E}[\Delta_{\mathcal{P}}^{(d)}(E_i,t) \mid E_i'] \\ &- \mathbb{E}[\Delta_{\mathcal{N}}^{(\text{other})}(E_i,t) \mid E_i']\Big]\end{aligned} \tag{14}$$

Therefore, the expected total performance change can be approximated as:

$$\begin{aligned}\mathbb{E}[C_i] &= \mathrm{AF}_i^{(d)} \left[k_1 \,\mathrm{EDMS}_i^{(d)} - k_2\,(1 - \mathrm{EDMS}_i^{(d)})\right] \\ &= \mathrm{AF}_i^{(d)} \left[(k_1 + k_2)\,\mathrm{EDMS}_i^{(d)} - k_2\right]\end{aligned} \tag{15}$$

Our objective is to select an expert set S that maximizes the expected total performance change after fine-tuning in the target d, i.e.,

$$\mathrm{S} = \arg\max_{|\mathrm{S}|=m} \sum_{i \in \mathrm{S}} \mathbb{E}[C_i]$$

. Since k_1 and k_2 are constants, they do not affect the ranking of expert selection. Thus, the optimization objective is equivalent to:

$$\begin{aligned}\mathrm{S} &= \arg\max_{|\mathrm{S}|=m} \sum_{i \in \mathrm{S}} \mathbb{E}[C_i] \\ &= \arg\max_{|\mathrm{S}|=m} \sum_{i \in \mathrm{S}} \mathrm{EDMRS}_i^{(d)}\end{aligned} \tag{16}$$

In other words, to maximize the total EDMRS score, the m experts with the highest EDMRS scores should be selected, which leads to our EDMRS strategy and demonstrates its optimality. □

A.5 Experiments Settings

In our experiments, GPT-4o is employed as the external large language model within the analytical framework, while Qwen3-30B-A3B-Instruct-2507 serves as the base model. For the comparative experiments on the MedMCQA dataset involving three strategies, the hyperparameter configurations are as follows: a batch size of 32 and a learning rate of 5e-5 are used. Due to differences in the number of parameters updated at each training step, the EDMRS and AF models are trained for 3 epochs, whereas the Full-parameter model is trained for 1 epoch. For the LegalBench dataset, we randomly

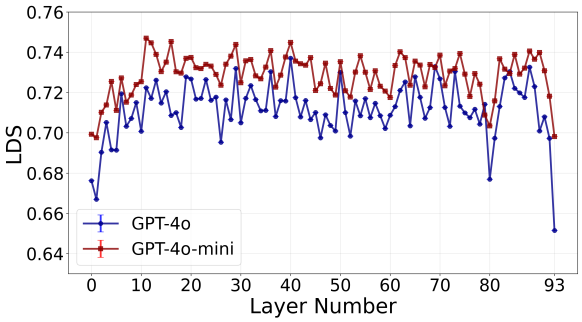


Figure 4: Analysis of metric stability under affordable external LLMs.

sample 20% of the data for training and testing, respectively. All methods utilize a batch size of 32, employ a cosine annealing learning rate schedule, with initial learning rates set to 1e-5 for AF and EDMRS, and 5e-6 for the Full model. A warmup ratio of 0.05 is applied, and the minimum learning rates are set to 1e-6 for AF and EDMRS, and 5e-7 for the Full model. All models are trained for 1 epoch. We report the maximum value across 5 random seeds. When applying the expert selection fine-tuning strategy, we select all layers of the model and only fine-tune 16 experts within each layer.

A.6 Stability of Metrics with Affordable External LLMs

To evaluate the stability of monosemanticity metrics when employing more cost-effective external large language models, we conducted experiments on Qwen3-235B-A22B using both GPT-4o and the smaller, lower-cost GPT-4o-mini to assess all layers. Comparative results are presented in Figure 4, where the relative variation trends demonstrate high consistency across different external LLMs, reflecting the robustness of monosemanticity prediction. Thus, even when shifting to a more economical external LLM, our scoring remains highly reliable.

A.7 Use of Ai Assistants

The present manuscript's Introduction and Conclusion underwent limited language refinement through a general-purpose large language model, strictly for improving grammar, wording, and readability. The model played no role in conceptualization, data or code generation, image production, mathematical derivation, data analysis, literature search, or citation management. Its use was confined to light textual polishing of pre-written content, with no confidential or external data disclosed. All AI-proposed changes were carefully examined and manually approved by the authors, who retain sole accountability for the scholarly content of the work.