Do Domain-specific Experts exist in MoE-based LLMs?

Giang Do* **Hung Le** **Truyen Tran**
Applied Artificial Intelligence Intiative (A2I2), Deakin University
{truong.do,thai.le,truyen.tran}@deakin.edu.au

Abstract

In the era of Large Language Models (LLMs), the Mixture of Experts (MoE) architecture has emerged as an effective approach for training extremely large models with improved computational efficiency. This success builds upon extensive prior research aimed at enhancing expert specialization in MoE-based LLMs. However, the nature of such specializations and how they can be systematically interpreted remain open research challenges. In this work, we investigate this gap by posing a fundamental question: *Do domain-specific experts exist in MoE-based LLMs?* To answer the question, we evaluate ten advanced MoE-based LLMs ranging from 3.8B to 120B parameters and provide empirical evidence for the existence of domain-specific experts. Building on this finding, we propose **Domain Steering Mixture of Experts (DSMoE)**, a training-free framework that introduces zero additional inference cost and outperforms both well-trained MoE-based LLMs and strong baselines, including Supervised Fine-Tuning (SFT). Experiments on four advanced open-source MoE-based LLMs across both target and non-target domains demonstrate that our method achieves strong performance and robust generalization without increasing inference cost or requiring additional retraining. Our implementation is publicly available at https://github.com/giangdip2410/Domain-specific-Experts.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success in Natural Language Processing (NLP) (Brown et al., 2020; Du et al., 2022; Fedus et al., 2022), Computer Vision (Riquelme et al., 2021a; Shen et al., 2023), and multimodal applications (Zhan et al., 2024; Li et al., 2025a). This progress is largely driven by scaling laws (Kaplan et al., 2020), which indicate that LLM performance strongly correlates with model scale. Mixture of Experts (MoE) architectures (Shazeer et al., 2017) are particularly well suited for scaling models under a fixed computational budget, making them an effective paradigm for building large-capacity models efficiently (Team, 2024; Dai et al., 2024).

The success of MoE-based LLMs is driven by extensive work on improving expert specialization to enable efficient training and fine-tuning (Krishnamurthy et al., 2023; Dai et al., 2024; Wang et al., 2024b). Yet, the nature and interpretability of such specializations remain open questions. In this work, we investigate this gap by posing a fundamental question: *Do domain-specific experts exist in MoE-based LLMs?* To tackle this problem, we conduct a systematic evaluation of recent open-source MoE-based LLMs across diverse parameter scales, ranging from large models such as GPT-OSS-120B (OpenAI et al., 2025) (120B parameters) to compact models such as Phi-tiny-MoE (Li et al., 2025b) (3.8B parameters).

Building on these findings, we introduce **Domain Steering Mixture of Experts (DSMoE)**, a training-free framework that guides MoE-based LLMs toward improved performance while incurring no additional inference cost and requiring no further retraining. We conduct extensive experiments on four state-of-the-art MoE-based LLMs, including GPT-OSS (OpenAI et al., 2025) and Qwen3MoE (Yang et al., 2025), across both target and non-target domains. Evaluations on challenging reasoning benchmarks such as MMLU-Pro (Wang et al., 2024a) and GPQA-Diamond (Rein et al., 2023) demonstrate that our method achieves strong performance while exhibiting robust generalization without increasing inference cost or requiring additional retraining. In summary, this paper makes three key contributions:

- We formulate the interpretability of MoE-based LLMs as the problem of verifying

*Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 2324–2340

Question	There exist real numbers x and y, both greater than 1, such that $\log_x(y^x) = \log_y(x^{4y}) = 10$. Find xy.
Baseline (GPT-OSS-20B)	*Response:* We need to solve for $x, y > 1$. [...] Equation (1): $y^x = x^{10} \implies be^a = 10a$. Equation (2): $x^{4y} = y^{10} \implies 4ae^b = 10b$. We can attempt to solve for a, b. From (I): $b = 10a/e^a$. Plug into (II): $4ae^{10a/e^a} = 10b$. **That seems messy.** Alternatively, divide (I) by (II): $e^{a-b} = 4a^2/b^2$. [...] We can attempt to solve for t numerically. Let's try $t = 1, t = 2, t = 0.5$... [loops through 20+ numerical guesses]. So left side negative until $t \approx 2.5$, right side positive. **So no crossing? So equality cannot hold.** **Predicted Answer:** 0 **(Incorrect)**
DSMoE (Ours)	*Response:* We have $x \ln y = 10 \ln x$ and $4y \ln x = 10 \ln y$. Let $u = \ln x, v = \ln y$. Equations: $e^u v = 10u$ and $4e^v u = 10v$. Dividing equations leads to $u - v = \ln 4 + 2 \ln u - 2 \ln v$. [...] Let's attempt to find integer solution for xy. Suppose $xy = 25$. We have $u + v = \ln 25 \approx 3.218$. Solving numerically for u with sum $s = \ln 25$: $e^u(s - u) = 10u$. We find $u \approx 0.4275$. Checking second equation: $4e^{s-u}u = 10(s - u)$. Left side: 27.904. Right side: 27.913. **Very close. So indeed u solves both. So $xy = 25$ is consistent.** **Predicted Answer:** 25 **(Correct)**

Table 1: Qualitative comparison on a challenging math problem - **AIME** (MAA Committees, n.d.) 2024. The baseline (GPT-OSS-20B) fails to resolve the system of equations, getting stuck in numerical approximation. In contrast, DSMoE successfully identifies the integer relationship $xy = 25$ through effective hypothesis testing.

domain-specific experts.

- We empirically evaluate ten advanced MoE-based LLMs across a wide range of model scales, providing evidence for the existence of domain-specific specialization.
- We propose **Domain Steering MoEs (DSMoE)**, a training-free framework that achieves strong performance and generalization without incurring additional inference or retraining costs.

2 Related work

Expert Specialization. Mixture of Experts (MoE) models (Jacobs et al., 1991; Jordan and Jacobs, 1994) have gained significant traction in large language models and have since been widely applied across domains such as natural language processing, computer vision, and speech recognition (Jiang et al., 2024; Zhou et al., 2022; Riquelme et al., 2021b). However, ensuring that experts acquire non-overlapping and specialized knowledge remains challenging (Dai et al., 2024; Chi et al., 2022). To address this challenge, prior work has followed two main research directions: (1) modifying the MoE architecture and (2) improving the routing mechanism. Following the first direction, DeepSeekMoE (Dai et al., 2024) promotes expert specialization through fine-grained expert segmentation and the introduction of shared experts, while μMoE (Oldfield et al., 2024) achieves specialization by performing implicit computation over prohibitively large weight tensors entirely in a factorized form. Along the second direction, various routing-based solutions have been proposed, including XMoE, which employs low-dimensional routing scores (Chi et al., 2022), and SMoE-dropout, which gradually activates a larger number of experts during training (Chen et al., 2023). Other approaches, such as StableMoE (Dai et al., 2022) and HyperRouter (Do et al., 2023), focus on improving router stability and robustness. Beyond architectural and routing modifications, some studies propose auxiliary loss functions to further enhance expert specialization in MoE-based LLMs (Do et al., 2024; Guo et al., 2025).

Explainable MoE. The demand for reliable and transparent model explanations is critical across many machine learning applications, for which Mixture of Experts (MoE) architectures are particularly well suited. Interpretable MoE (IME) (Ismail et al., 2023) integrates MoE with deep neural networks (DNNs) to achieve both strong predictive performance and enhanced interpretabil-

ity. Following a similar line of research, SMoE-VAE (Nikolic et al., 2025) combines MoE with variational autoencoders to improve expert-level interpretability in an unsupervised setting. More recently, MoE Lens (Chaudhari et al., 2025) analyzes domain-specific routing patterns and demonstrates that MoE models predominantly rely on a small subset of specialized experts, with the top-weighted expert's output closely approximating the full ensemble prediction.

Steering LLMs. Controlling the behavior of large language models (LLMs) through direct intervention on internal activations has emerged as a promising research direction. Prior work has proposed various activation-steering methods that modify model behavior by injecting steering signals into intermediate representations (Turner et al., 2024; Rimsky et al., 2024). Most existing approaches apply a steering vector to the model activations at specific layers and token positions during inference; however, such methods typically rely on locally learned steering vectors, which limits their ability to generalize across multiple domains (Wang et al., 2025a). RICE (Wang et al., 2025a) recently introduces a steering method that focuses on *thinking experts*. However, this approach is currently limited to extremely large reasoning models, such as DeepSeek-R1 (DeepSeek-AI et al., 2025) and Qwen3-235B (Team, 2025). Moreover, RICE targets thinking experts that exhibit substantial variability across samples or domains, which can hinder the method's generalization capability. In contrast, our work addresses a fundamental question: *Do domain-specific experts exist in MoE-based LLMs?* To this end, we conduct a systematic evaluation of recent open-source MoE-based LLMs across a wide range of parameter scales, such as GPT-OSS-120B (OpenAI et al., 2025) or Phi-tiny-MoE (Li et al., 2025b). Based on these conclusions, we propose **Domain Steering Mixture of Experts (DSMoE)**, a training-free framework that introduces zero additional inference cost and consistently outperforms well-trained MoE-based LLMs as well as strong baselines, including supervised fine-tuning (SFT).

3 Methodology

3.1 Preliminaries

MoE-based LLM Layer. An MoE-based LLM layer replaces the standard Feedforward Network (FFN) with a Mixture of Experts module consisting of N experts. Given an input $\boldsymbol{x} \in \mathbb{R}^d$, the layer computes a weighted aggregation of k active experts:

$$f^{\mathrm{MoE}}(\boldsymbol{x}) = \sum_{i \in \mathcal{K}} s_i(\boldsymbol{x})\, \mathrm{FFN}_i(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{W}_e \in \mathbb{R}^{N \times d}$ is the router embedding and $\mathcal{K} \subset \{1, \ldots, N\}$ is the set of k selected indices. The gating weights $\boldsymbol{s}(\boldsymbol{x})$ are derived from the router logits $\boldsymbol{l} = \boldsymbol{W}_e \boldsymbol{x}$. Depending on the architecture, sparsity is enforced either *after* softmax (standard) or *before* softmax (masked). Letting $\sigma(\cdot)$ denote the softmax function:

$$\boldsymbol{s}(\boldsymbol{x}) = \begin{cases} \mathrm{TopK}(\sigma(\boldsymbol{l}), k) & \text{(Post-Softmax)} \\ \sigma(\mathrm{TopK}_{\mathrm{mask}}(\boldsymbol{l}, k)) & \text{(Pre-Softmax)} \end{cases} \tag{2}$$

where $\mathrm{TopK}_{\mathrm{mask}}$ sets the logits of non-selected experts to $-\infty$ prior to normalization. Each expert FFN_i is a multi-layer perceptron.

3.2 Domain-specific Token

Definition 3.1 (Common Token) *Let $\mathcal{D}$ denote a specific domain and $S = (s_1, s_2, \ldots, s_T)$ be a sequence of tokens such that $S \in \mathcal{D}$. Let $\mathcal{M}$ be a pre-trained Large Language Model and $\mathcal{T}(\cdot, \mathcal{M})$ be the task evaluation metric.*

A token s_i is defined as a Common Token if its removal results in a negligible change in the task metric, bounded by a small threshold ϵ:

$$\left|\mathcal{T}(S, \mathcal{M}) - \mathcal{T}(S_{\setminus\{s_i\}}, \mathcal{M})\right| < \epsilon \tag{3}$$

where $S_{\setminus\{s_i\}}$ denotes the sequence S excluding token s_i, and $\epsilon \approx 0$ is a small scalar.

Definition 3.2 (Domain-specific Token) *Following the notation in Definition 3.1, a token s_i is defined as a Domain-specific Token if its removal causes a significant degradation in the task metric, exceeding the threshold ϵ:*

$$\left|\mathcal{T}(S, \mathcal{M}) - \mathcal{T}(S_{\setminus\{s_i\}}, \mathcal{M})\right| \geq \epsilon \tag{4}$$

This inequality implies that s_i carries information crucial to the performance of model $\mathcal{M}$ on domain $\mathcal{D}$.

Remark: Unless otherwise specified, $\mathcal{T}(\cdot, \mathcal{M})$ denotes the cross-entropy loss, as this is the standard objective function for next-token prediction tasks in Large Language Models.

A direct approach to identifying domain-specific tokens satisfying Definition 3.2 is Leave-One-Out

(LOO) cross-validation (Cawley, 2006). However, this method is computationally prohibitive for long sequences, as it scales with complexity $\mathcal{O}(N^2)$. To overcome this limitation, we leverage gradient-based attribution methods to approximate token importance (Shrikumar et al., 2019; Ancona et al., 2018).

Specifically, let $\mathbf{e}_i \in \mathbb{R}^d$ denote the input embedding vector for token s_i, and let $\mathcal{L}$ be the task loss. We calculate the ranking score r_i for each token as the L_2 norm of the element-wise product between the embedding and its gradient:

$$r_i = \|\mathbf{e}_i \odot \nabla_{\mathbf{e}_i} \mathcal{L}\|_2 \tag{5}$$

where $\odot$ denotes the Hadamard product and $\nabla_{\mathbf{e}_i} \mathcal{L}$ is the gradient of the loss with respect to the embedding vector. High values of r_i indicate tokens that significantly influence the model's output.

Definition 3.3 (Domain-specific Threshold) *Let $r = (r_1, r_2, \ldots, r_T)$ be the vector of token importance scores for sequence S. For a chosen domain-specific level $p \in [0, 1]$ (representing the proportion of tokens considered domain-specific), we define the threshold t_p as the value satisfying the empirical quantile condition:*

$$\frac{1}{T} \sum_{i=1}^{T} \mathbb{I}(|r_i| \leq t_p) = 1 - p \tag{6}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The token classification function $\mathcal{C}_{t_p} : \mathbb{R}^T \to \{\text{Common, Specific}\}^T$ is defined as:

$$\mathcal{C}_{t_p}(r)_i = \begin{cases} \textit{Common Token} & \textit{if } |r_i| \leq t_p \\ \textit{Domain-specific Token} & \textit{otherwise} \end{cases} \tag{7}$$

Remark: The hyperparameter p is selected empirically. We find that values in the range $(0.15, 0.5)$ are effective, a setting consistent with the 15% masking ratio used in the BERT pre-training objective (Devlin et al., 2019).

3.3 Domain-specific Expert

We hypothesize the existence of *domain-specific experts* within the Mixture-of-Experts (MoE) architecture. These experts are characterized by a dual property: they are frequently activated within a specific domain, and when activated, they exhibit a strong preference for processing domain-specific tokens rather than common tokens.

Definition 3.4 (Domain-specific Expert) *Let S be a sequence of tokens from domain $\mathcal{D}$, partitioned into a set of domain-specific tokens $\mathcal{S}$ and common tokens $\mathcal{C}$ (as per Definition 3.3). Let $\mathcal{E} = \{e_1, \ldots, e_N\}$ be the set of N experts in the model.*

For each expert e_j, we calculate the domain-specific score $g(e_j)$ as:

$$g(e_j) = P(e_j|\mathcal{D}) \cdot [P(s \in \mathcal{S}|e_j) - P(s \in \mathcal{C}|e_j)] \tag{8}$$

where $P(e_j|\mathcal{D})$ is the activation frequency of e_j on domain $\mathcal{D}$, and the conditional probabilities represent the expert's token preference.

Let K be a hyperparameter denoting the target number of domain-specific experts (i.e., top-K). We define the selection threshold γ_K such that:

$$\sum_{j=1}^{N} \mathbb{I}(g(e_j) \geq \gamma_K) = K \tag{9}$$

Accordingly, the set of Domain-specific Experts $\mathcal{E}^* \subset \mathcal{E}$ is defined as the subset of experts satisfying this condition:

$$\mathcal{E}^* = \{e_j \in \mathcal{E} \mid g(e_j) \geq \gamma_K\} \tag{10}$$

Remark: The score $g(e_j)$ effectively balances two factors: the expert's global relevance to the domain (represented by $P(e_j|\mathcal{D})$) and its specialization level (represented by the difference in conditional probabilities). This penalizes experts that are active frequently but only process common, non-informative tokens.

3.4 Domain Steering Mixture of Experts (DSMoE)

Building upon the identification of domain-specific experts in Definition 3.4, we propose **Domain Steering Mixture of Experts (DSMoE)**. DSMoE is a training-free inference framework designed to adapt generic MoE-based Large Language Models to specific target domains by dynamically modulating the router's behavior.

Formally, let $\mathcal{E}^*$ denote the set of identified domain-specific experts. For a given input token, let w_j represent the original scalar weight (or logit) assigned to expert e_j by the router. We introduce a *steering coefficient* $\alpha \in (0, 100)$ to amplify the contribution of domain-specific experts.

The steered routing weights $\tilde{w}_j$ are computed as follows:

$$\tilde{w}_j = \begin{cases} \alpha \cdot w_j & \text{if } e_j \in \mathcal{E}^* \\ w_j & \text{otherwise} \end{cases} \tag{11}$$

After applying this steering transformation, the weights are typically re-normalized (e.g., via a Softmax function) to ensure a valid probability distribution for expert selection. This mechanism effectively biases the model's computation path towards experts specialized for the target domain without requiring parameter updates.

4 Experiments

We design our experiments to investigate the following four research questions: **RQ1 (Existence)** asks if domain-specific experts exist in MoE-based LLMs; **RQ2 (Effectiveness)** evaluates how effective the proposed DSMoE framework is on target domains; **RQ3 (Generalization)** examines if DSMoE maintains robust performance on non-target domains; and **RQ4 (Efficiency)** analyzes the computational cost of DSMoE compared to baselines, particularly regarding inference overhead.

4.1 Experimental Settings

MoE-based LLMs. To validate our hypothesis, we evaluate our method across a diverse set of state-of-the-art Mixture-of-Experts (MoE) Large Language Models, ranging in scale from 3.8B to 120B parameters. Specifically, our experimental suite includes PhiMoE-Tiny (Li et al., 2025b), OLMoE (Muennighoff et al., 2024), Qwen1.5-MoE (Team, 2024), and DeepSeek-MoE (Dai et al., 2024). We also include recent advanced models such as GPT-OSS-20B and GPT-OSS-120B (OpenAI et al., 2025), ERNIE-4.5 (Baidu-ERNIE-Team, 2025), and the Qwen3-MoE series (Instruct, Think, and Next variants) (Team, 2025). Detailed architectural specifications for all utilized models are provided in Table 7.

Baselines. Since our proposed DSMoE is a training-free framework compatible with any off-the-shelf MoE-based LLMs, our main baseline is the Original MoE-based models. We also compare our approach against RICE (Wang et al., 2025b), a recent state-of-the-art method for steering MoE-based reasoning models. To provide a comprehensive evaluation, we further benchmark against Supervised Fine-Tuning (SFT) using LoRA (Hu et al., 2021), which serves as the standard paradigm for domain adaptation. Throughout this paper, *SFT* refers to fine-tuning using LoRA, with trainable parameters comprising 2.7% of the total model parameters, unless stated otherwise.

Benchmarks. To demonstrate the efficacy of DSMoE, we evaluate our method on three challenging benchmarks designed to assess deep domain understanding and complex reasoning. We first utilize **MMLU-Pro** (Wang et al., 2024a), a robust benchmark that introduces reasoning-intensive questions with a ten-option choice set to minimize random guessing. Furthermore, we test on **GPQA Diamond** (Rein et al., 2023), a graduate-level dataset of such extreme difficulty that domain experts (PhDs) achieve only $\approx 65\%$ accuracy. Finally, we evaluate advanced mathematical proficiency using **AIME** (MAA Committees, n.d.), a collection of problems from the American Invitational Mathematics Examination known for requiring multi-step reasoning.

4.2 Domain-specific Experts Testing

To verify that the domain-specific experts defined in Section 3.4 are genuinely specialized, we conduct a performance evaluation on the mathematics domain using ten advanced MoE-based LLMs. For each model, we sample 10% of mathematics questions from the MMLU-Pro dataset. The evaluation follows the procedure described in Section 3, consisting of three steps: (1) identifying domain-specific tokens as defined in Definition 3.2; (2) computing expert ranking scores according to Equation 8; and (3) applying DSMoE with $K = 1$ by activating the expert with the highest ranking score.

For Step (1), to ensure a fair evaluation, we identify domain-specific tokens using only the question tokens, excluding answer tokens. We visualize the evaluation results by comparing DSMoE predictions with the ground-truth labels of the MMLU-Pro mathematics benchmark, as shown in Figure 1. The results demonstrate that at least one domain-specific expert exists for the mathematics domain in all ten evaluated MoE-based LLMs. These findings support our hypothesis that MoE-based LLMs inherently contain domain-specific experts. Moreover, an interesting observation is that steering a single expert in MoE-based LLMs can significantly improve performance over the base model, with gains ranging from **3%** to **45%**.

4.3 Target Domains Evaluation

Conventional steering methods, such as RICE (Wang et al., 2025a), require rerunning the steering procedure separately for each domain and each dataset. In contrast, DSMoE is built upon domain-specific representations

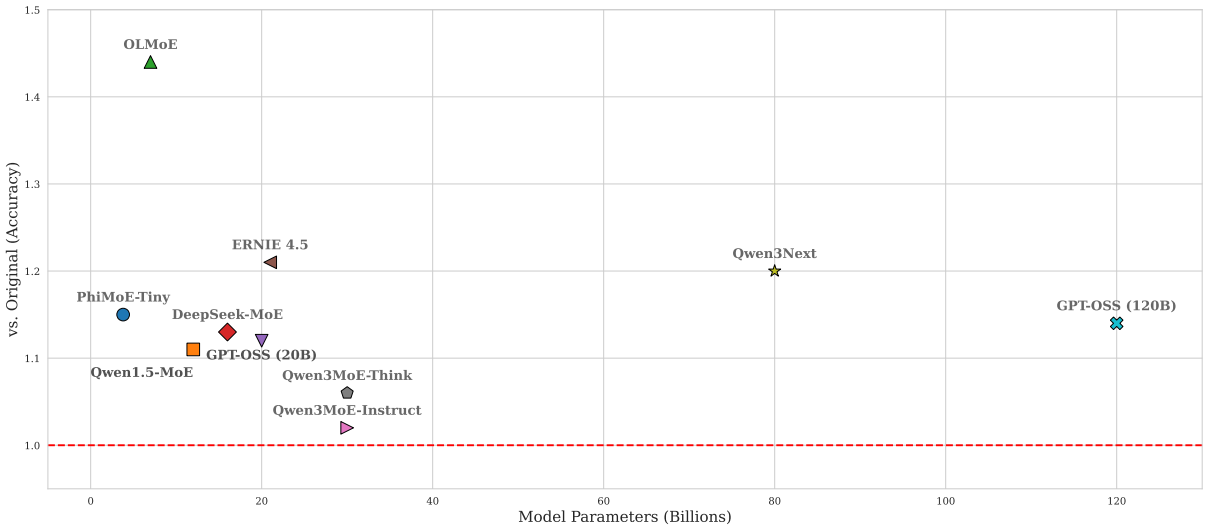


Figure 1: Evaluation of Domain-specific Experts across ten state-of-the-art MoE LLMs on the MMLU-Pro Math domain. Experiments were conducted with $K = 1$ (number of specific experts) and $\alpha = 3.0$ (steering coefficient). All models exhibit performance improvements compared to their original baselines, providing empirical evidence for the existence of domain-specific experts within MoE architectures. Best viewed in color.

and requires identifying domain-specific tokens (Definition 3.2) and domain-specific experts (Definition 3.4) only once. Since the MMLU-Pro dataset supports 14 domains, we adopt it as the target dataset for identifying domain-specific tokens and experts. In this work, we focus on four target domains (Math, Biology, Physics, and Chemistry); however, DSMoE can be straightforwardly extended to other domains supported by MMLU-Pro.

For each target domain, we use only question texts (without labels) from MMLU-Pro to identify domain-specific tokens and domain-specific experts. Empirically, we find that $p \in (0.15, 0.5)$ and K set to approximately 1% of the total number of experts yield strong performance. Notably, smaller values of p impose a stricter criterion for domain specificity, resulting in fewer selected experts but higher confidence in their domain specialization. Table 2 presents the comparative results across four target domains on the MMLU-Pro benchmark. Overall, DSMoE consistently outperforms both the original MoE baselines and the advanced steering method, RICE, across all evaluated models. Specifically, DSMoE yields average absolute improvements of **+1.5**, **+14.5**, **+3.6**, and **+3.7** percentage points for Qwen3-30B-Instruct, GPT-OSS-120B, Qwen3-30B-Thinking, and GPT-OSS-20B, respectively.

Our analysis highlights significant limitations in baseline methods. The results indicate that RICE, which relies heavily on thinking tokens, struggles to generalize to standard (non-reasoning) models. Furthermore, DSMoE frequently outperforms Supervised Fine-Tuning (SFT). While SFT requires updating approximately 2.7% of the model's parameters and is inherently data-intensive, DSMoE achieves superior performance without any parameter updates. Notably, our method achieves substantial gains in challenging domains such as Mathematics and Chemistry, recording improvements of up to **+29.1** points over the original model. These findings demonstrate that selectively activating domain-specific experts is a highly efficient strategy for enhancing model performance without the computational and data costs of fine-tuning.

4.4 Non-target Domains Evaluation

Generalizability Evaluation.; To demonstrate the generalizability of DSMoE, we evaluate its performance on two independent benchmarks using domain-specific experts identified via the MMLU-Pro dataset. First, Table 3 reports results on the GPQA Diamond benchmark across Biology, Physics, and Chemistry. DSMoE consistently outperforms the original MoE baselines across all evaluated models, yielding average gains ranging from **+4.8** to **+27.1** percentage points. To further assess performance on extremely challenging reasoning tasks, we evaluate DSMoE on the American Invitational Mathematics Examination (AIME) (MAA Committees, n.d.). As shown in Table 4, DSMoE

Domain	Orig.	RICE	SFT	DSMoE	+/-
Qwen3-30B-Instruct					
Math	85.0	85.9	85.4	**86.3**	+1.3
Biology	87.6	87.0	87.9	**88.6**	+1.0
Physics	78.1	79.1	78.8	**80.0**	+1.8
Chemistry	76.2	77.7	77.1	**78.3**	+2.0
Average	81.7	82.4	82.3	**83.3**	+1.5
GPT-OSS-120B					
Math	74.0	82.8	77.4	**87.5**	+13.5
Biology	72.8	85.8	86.8	**88.1**	+15.3
Physics	82.4	81.8	80.9	**82.6**	+0.2
Chemistry	50.2	78.0	**79.8**	79.2	+29.1
Average	69.8	82.1	81.2	**84.4**	+14.5
Qwen3-30B-Thinking					
Math	75.3	74.5	72.8	**78.3**	+3.1
Biology	73.9	74.3	73.2	**75.2**	+1.3
Physics	58.7	59.4	61.8	**65.8**	+7.1
Chemistry	56.5	**59.3**	58.7	**59.3**	+2.8
Average	66.1	66.9	66.6	**69.6**	+3.6
GPT-OSS-20B					
Math	66.5	64.7	62.6	**74.5**	+8.1
Biology	76.4	73.1	73.9	**78.7**	+2.2
Physics	62.0	57.3	58.2	**65.7**	+3.8
Chemistry	48.9	48.9	47.3	**49.6**	+0.7
Average	63.4	61.0	60.5	**67.1**	+3.7

Table 2: Accuracy (%) on the MMLU-Pro benchmark across different domains. Best results per row are highlighted in **bold**. The Δ column indicates the absolute improvement (in percentage points) of DSMoE over the original MoE-based LLMs.

surpasses all baselines on both the 2024 and 2025 datasets using Qwen3-30B-Instruct and GPT-OSS-20B. Remarkably, DSMoE achieves improvements ranging from **+12.3** to **+27.3** percentage points over the baseline. These results confirm that DSMoE effectively generalizes across datasets of varying difficulty levels within a specific domain, maintaining superior performance even on competition-grade problems.

Notably, DSMoE exhibits robust transfer capabilities without systematic degradation on these unseen tasks. In many instances, it surpasses both RICE (a training-free steering baseline) and SFT (a fine-tuning method). The improvements are particularly pronounced for smaller architectures, such as GPT-OSS-20B, suggesting that selectively activating domain-relevant experts enhances fundamental scientific reasoning beyond the specific dataset used for identification. These results indicate that DSMoE effectively preserves, and often significantly enhances, cross-domain generalization while applying targeted expert steering.

Domain	Orig.	RICE	SFT	DSMoE	+/-
Qwen3-30B-Instruct					
Biology	68.4	47.4	63.2	**73.7**	+5.3
Physics	70.9	70.9	74.4	**76.7**	+5.8
Chemistry	53.8	45.2	41.9	**58.1**	+4.3
Average	64.4	54.5	59.8	**69.5**	+5.1
GPT-OSS-120B					
Biology	58.0	63.2	**73.7**	68.4	+10.4
Physics	88.4	86.1	82.6	**89.5**	+1.2
Chemistry	50.5	43.0	40.9	**59.1**	+8.6
Average	65.6	64.1	65.7	**72.4**	+6.7
Qwen3-30B-Thinking					
Biology	52.6	42.1	21.1	**57.9**	+5.3
Physics	55.8	46.5	54.7	**60.5**	+4.7
Chemistry	47.3	20.4	26.9	**51.6**	+4.3
Average	51.9	36.4	34.2	**56.7**	+4.8
GPT-OSS-20B					
Biology	47.4	57.9	63.2	**84.2**	+36.8
Physics	62.8	68.6	66.3	**80.2**	+17.4
Chemistry	47.3	34.4	41.9	**74.2**	+26.9
Average	52.5	53.6	57.1	**79.5**	+27.1

Table 3: Performance (accuracy) comparison on the GPQA Diamond dataset across science domains. All values are reported in percentage (%). Best results are **bolded**.

Dataset	Orig.	RICE	SFT	DSMoE	+/-
Qwen3-30B-Instruct					
AIME 24	56.7	63.3	63.3	**70.0**	+13.3
AIME 25	46.7	46.7	50.0	**60.0**	+13.3
GPT-OSS-20B					
AIME 24	50.0	56.7	56.7	**77.3**	+27.3
AIME 25	66.3	46.7	60.0	**78.6**	+12.3

Table 4: Accuracy comparison on Math benchmarks (AIME 24 & 25). All values are in percentages (%). Best results are highlighted in **bold**.

4.5 Cost Analysis

One-time Cost. We analyze the computational cost of identifying domain-specific experts. For DSMoE, the one-time identification cost scales linearly with the number of samples in a domain, yielding a time complexity of $O(L)$ forward passes, where L denotes the number of domain-specific

samples. In contrast, the RICE baseline incurs a substantially higher cost of $O(L \times M)$ forward passes, where M is the number of generated thinking tokens per sample. Since $M \gg 1$ in practice, this results in orders-of-magnitude higher computational overhead. Consequently, DSMoE is significantly more efficient than RICE for one-time expert identification.

Inference Cost. DSMoE uses a router weight steering approach, where the steered expert weights are computed once and stored for future inference. As a result, DSMoE maintains the same inference cost as the original MoE-based LLMs. Interestingly, DSMoE can produce answers using fewer thinking tokens, as illustrated in Table 1, making MoE-based models more efficient during inference. RICE applies steering to the router scores on a per-sample basis, which cannot be precomputed. Consequently, RICE incurs higher inference cost compared to the original models.

4.6 Ablation Studies

To determine the optimal number of domain-specific experts, we conduct an ablation study on the number of activated domain-specific experts (K) using GPT-OSS-20B on the Biology domain, with results reported in Table 5. When $K = 0$, corresponding to the original model without domain-specific routing, performance is lower than the best DSMoE configurations. As K increases, performance initially improves and reaches its peak at $K = 20$, indicating an effective balance between expert specialization and routing diversity. Beyond this point, further increases in K yield diminishing returns. In practice, we find that setting the number of domain-specific experts to approximately 1% of the total expert count often serves as an optimal hyperparameter.

Table 6 analyzes the effect of the steering coefficient (α) on DSMoE performance using GPT-OSS-20B in the Biology domain with $K = 20$. Compared to the original MoE-based LLMs, moderate steering improves performance, with optimal results achieved at $\alpha = 5.0$. As α increases from lower values, performance gradually improves and reaches its peak at $\alpha = 5.0$, before declining at higher values. This trend suggests that excessively small or large steering coefficients can lead to suboptimal routing behavior, either by providing insufficient guidance or by overly constraining expert selection. In practice, we observe that steering coefficients in the range of $[2.0, 5.0]$ consistently yield positive results across different configurations.

K	Domain	Method	GPT-OSS-20B
0	Biology	Original	76.4
5		DSMoE	73.6
10			73.3
20			**78.7**
30			77.4
50			74.1

Table 5: Ablation study on the number of domain-specific experts (K) on the MMLU-Pro dataset. The horizontal line separates the baseline (Original) from the steering method (DSMoE). The best performance (accuracy) is achieved at $K = 20$.

α	Domain	Method	GPT-OSS-20B
–	Biology	Original	76.4
0.1		DSMoE	77.0
0.5			76.3
5.0			**78.7**
10.0			77.0
50.0			68.0

Table 6: Ablation study on the steering coefficient (α) on the MMLU-Pro dataset. The Original method represents the baseline without steering. The best performance (accuracy) is observed at $\alpha = 5.0$.

5 Conclusion

This paper addressed the existence of domain-specific experts in MoE-based LLMs. Following a comprehensive analysis of models up to 120B parameters, we confirmed that distinct experts align with specific domains. We subsequently proposed **Domain Steering Mixture of Experts (DSMoE)**, a training-free, zero-overhead framework. Extensive experiments confirm that DSMoE surpasses strong baselines, including Supervised Fine-Tuning, delivering consistent performance gains across both target and non-target domains. These findings suggest that exploiting intrinsic expert specialization is a highly efficient alternative to traditional fine-tuning.

Limitations

This study focuses on enhancing the efficiency and effectiveness of MoE-based Large Language Models through a training-free approach. While the

results are promising, our experiments were constrained by computational resources, limiting the evaluation to medium-scale datasets and models up to GPT-OSS-120B. Future work should assess the scalability of DSMoE beyond 400B parameters and benchmark it against other Large Reasoning Models, such as DeepSeek-R1.

Ethics Statement

Despite the encouraging results, inference with large-scale LLMs remains highly resource-intensive, necessitating careful management of computational costs and environmental impact. Additionally, our study relies on web-sourced data, which may contain inherent gender and racial biases; future work should explore mitigation strategies to address these concerns. Finally, while this work represents a meaningful step toward advancing LLM development, it also underscores the importance of implementing robust safeguards to prevent potential misuse in harmful applications.

References

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for deep neural networks. *Preprint*, arXiv:1711.06104.

Baidu-ERNIE-Team. 2025. Ernie 4.5 technical report. https://ernie.baidu.com/blog/publication/ERNIE_Technical_Report.pdf.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

G.C. Cawley. 2006. Leave-one-out cross-validation based model selection criteria for weighted ls-svms. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1661–1668.

Marmik Chaudhari, Idhant Gulati, Nishkal Hundia, Pranav Karra, and Shivam Raval. 2025. Moe lens - an expert is all you need. In *Sparsity in LLMs (SLLM): Deep Dive into Mixture of Experts, Quantization, Hardware, and Inference*.

Tianlong Chen, Zhenyu Zhang, Ajay Jaiswal, Shiwei Liu, and Zhangyang Wang. 2023. Sparse moe as the new dropout: Scaling dense and self-slimmable transformers. *Preprint*, arXiv:2303.01610.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2022. On the representation collapse of sparse mixture of experts. *Preprint*, arXiv:2204.09179.

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. *Preprint*, arXiv:2401.06066.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Stablemoe: Stable routing strategy for mixture of experts. *Preprint*, arXiv:2204.08396.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Giang Do, Hung Le, and Truyen Tran. 2024. Simsmoe: Solving representational collapse via similarity measure. *Preprint*, arXiv:2406.15883.

Giang Do, Khiem Le, Quang Pham, TrungTin Nguyen, Thanh-Nam Doan, Bint T. Nguyen, Chenghao Liu, Savitha Ramasamy, Xiaoli Li, and Steven Hoi. 2023. Hyperrouter: Towards efficient training and inference of sparse mixture of experts. *Preprint*, arXiv:2312.07035.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, and 8 others. 2022. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR.

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.

Hongcan Guo, Haolang Lu, Guoshun Nan, Bolun Chu, Jialin Zhuang, Yuan Yang, Wenhao Che, Sicong Leng, Qimei Cui, and Xudong Jiang. 2025. Advancing expert specialization for better moe. *Preprint*, arXiv:2505.22323.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Aya Abdelsalam Ismail, Sercan Ö. Arik, Jinsung Yoon, Ankur Taly, Soheil Feizi, and Tomas Pfister. 2023. Interpretable mixture of experts. *Preprint*, arXiv:2206.02107.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

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, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Michael Jordan and Robert Jacobs. 1994. Hierarchical mixtures of experts and the. *Neural computation*, 6:181–.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Yamuna Krishnamurthy, Chris Watkins, and Thomas Gaertner. 2023. Improving expert specialization in mixture of experts. *Preprint*, arXiv:2302.14703.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yunxin Li, Shenyuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. 2025a. Uni-moe: Scaling unified multimodal llms with mixture of experts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3424–3439.

Zichong Li, Chen Liang, Zixuan Zhang, Ilgee Hong, Young Jin Kim, Weizhu Chen, and Tuo Zhao. 2025b. Slimmoe: Structured compression of large moe models via expert slimming and distillation. *Preprint*, arXiv:2506.18349.

MAA Committees. n.d. AIME problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, and 5 others. 2024. Olmoe: Open mixture-of-experts language models. *Preprint*, arXiv:2409.02060.

Strahinja Nikolic, Ilker Oguz, and Demetri Psaltis. 2025. Exploring expert specialization through unsupervised training in sparse mixture of experts. *Preprint*, arXiv:2509.10025.

James Oldfield, Markos Georgopoulos, Grigorios G. Chrysos, Christos Tzelepis, Yannis Panagakis, Mihalis A. Nicolaou, Jiankang Deng, and Ioannis Patras. 2024. Multilinear mixture of experts: Scalable expert specialization through factorization. In *Advances in Neural Information Processing Systems*, volume 37, pages 53022–53063. Curran Associates, Inc.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. 2021a. Scaling vision with sparse mixture of experts. *Preprint*, arXiv:2106.05974.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. 2021b. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems*, volume 34, pages 8583–8595. Curran Associates, Inc.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Sheng Shen, Zhewei Yao, Chunyuan Li, Trevor Darrell, Kurt Keutzer, and Yuxiong He. 2023. Scaling vision-language models with sparse mixture of experts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11329–11344, Singapore. Association for Computational Linguistics.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2019. Learning important features through propagating activation differences. *Preprint*, arXiv:1704.02685.

Qwen Team. 2024. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters".

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *Preprint*, arXiv:2308.10248.

Mengru Wang, Xingyu Chen, Yue Wang, Zhiwei He, Jiahao Xu, Tian Liang, Qiuzhi Liu, Yunzhi Yao, Wenxuan Wang, Ruotian Ma, Haitao Mi, Ningyu Zhang, Zhaopeng Tu, Xiaolong Li, and Dong Yu. 2025a. Two experts are all you need for steering thinking: Reinforcing cognitive effort in moe reasoning models without additional training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Mengru Wang, Xingyu Chen, Yue Wang, Zhiwei He, Jiahao Xu, Tian Liang, Qiuzhi Liu, Yunzhi Yao, Wenxuan Wang, Ruotian Ma, Haitao Mi, Ningyu Zhang, Zhaopeng Tu, Xiaolong Li, and Dong Yu. 2025b. Two experts are all you need for steering thinking: Reinforcing cognitive effort in moe reasoning models without additional training. *Preprint*, arXiv:2505.14681.

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

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, Hang Yan, Jie Fu, Tao Gui, Tianxiang Sun, Yu-Gang Jiang, and Xipeng Qiu. 2024. AnyGPT: Unified multimodal LLM with discrete sequence modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9637–9662, Bangkok, Thailand. Association for Computational Linguistics.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, zhifeng Chen, Quoc V Le, and James Laudon. 2022. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, pages 7103–7114. Curran Associates, Inc.

A Appendix

This document is organized as follows: Appendix A.1 illustrates some further analysis of DSMoE. Appendix A.2 presents supplementary benchmarks descriptions, Appendix A.3 consists of additional experiments, and Appendix A.4 describes the implementation details in full.

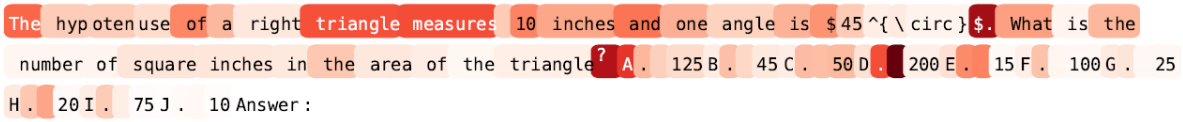
A.1 Domain-specific Experts Analysis

Figure 2 illustrates token ranking scores for five representative samples from the MMLU Mathematics domain evaluated with GPT-OSS-20B. The results show a highly non-uniform distribution of token importance across the input sequence, where tokens corresponding to mathematical expressions and key terms consistently receive markedly higher scores.

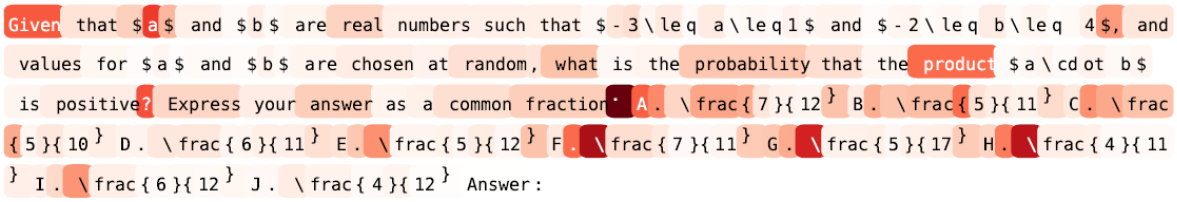
Figure 3 to Figure 6 present heatmap visualizations of domain-specific expert scores for four MoE-based LLMs on the Mathematics domain. Across all models, we observe that domain-specific experts are not uniformly distributed; instead, certain layers exhibit clusters of highly specialized experts (indicated by higher magnitudes), while others contain more domain-agnostic experts. This non-uniform distribution provides empirical evidence supporting the existence of domain-specific experts and motivates our proposed steering approach.

A.2 Benchmarks Descriptions

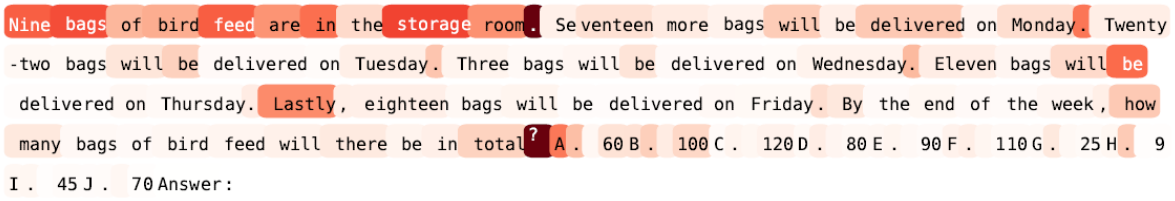
Table 7 summarizes the architectural specifications of the ten MoE-based LLMs used in our experi-



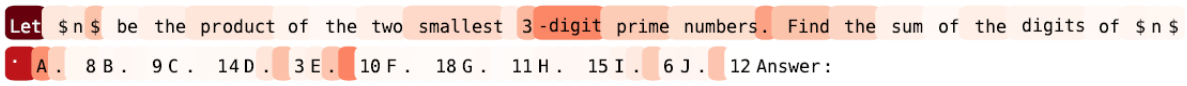
(a) Sample 1



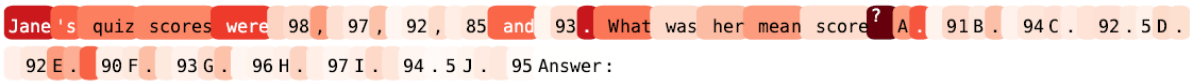
(b) Sample 2



(c) Sample 3



(d) Sample 4



(e) Sample 5

Figure 2: Token ranking scores across five representative samples from the MMLU-Pro, mathematics domain for GPT-OSS-20B. Each row displays the importance distribution of tokens within a single question, where higher scores indicate greater contribution to model predictions.

ments. The models span a wide range of scales, from 3.8B to 120B total parameters, with activated parameters ranging from 1.0B to 5.1B per token. The number of experts varies significantly across architectures, ranging from 16 (PhiMoE-Tiny) to 512 (Qwen3-Next), with Top-K routing selecting between 2 and 10 experts per token.

Table 8 summarizes the evaluation benchmarks used in our experiments. We assess model performance across three challenging benchmarks: MMLU-Pro (12K multi-domain questions), GPQA Diamond (198 PhD-level science questions), and AIME (30 competition-level mathematics problems).

A.3 Additional Experiments

A.3.1 Hyperparameter Consistency

To assess the generalizability of DSMoE, we adopt a strict hyperparameter transfer protocol. Specifically, the hyperparameter settings (K, α) obtained from Table 2 on the MMLU-Pro benchmark are directly applied to all corresponding domain-specific datasets reported in Tables 3 and 4, without any additional tuning. This ensures hyperparameter consistency and highlights the robustness and transferability of DSMoE.

The detailed hyperparameter values used for each model and domain are reported in Tables A.3.1

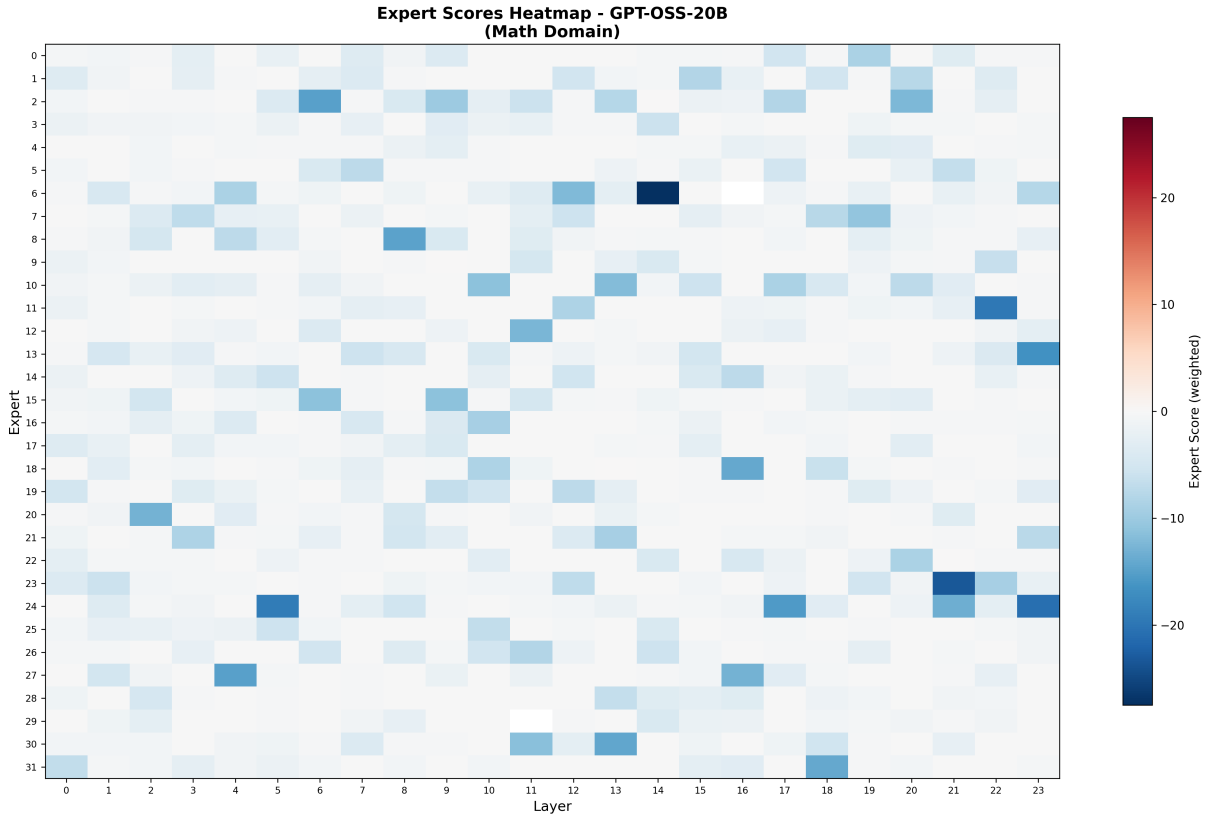


Figure 3: Domain-specific expert scores for **GPT-OSS-20B** on the Mathematics domain. Higher magnitudes indicate stronger domain specialization. Best viewed in color.

and A.3.1. Notably, the configurations in Table A.3.1 are identical to those in Table A.3.1, confirming that no dataset-specific re-tuning is performed when transferring to GPQA Diamond and AIME24–25.

MMLU-Pro

Model	Math		Biology		Chemistry		Physics	
	K	α	K	α	K	α	K	α
Qwen3-30B-Instruct	60	5.0	60	2.0	60	3.0	60	5.0
GPT-OSS-120B	90	3.0	90	3.0	90	2.0	90	3.0
Qwen3-30B-Thinking	60	5.0	60	6.0	60	6.0	60	5.0
GPT-OSS-20B	20	6.0	20	5.0	20	5.0	20	6.0

GPQA Diamond + AIME24–25

Model	Math		Biology		Chemistry		Physics	
	K	α	K	α	K	α	K	α
Qwen3-30B-Instruct	60	5.0	60	2.0	60	3.0	60	5.0
GPT-OSS-120B	90	3.0	90	3.0	90	2.0	90	3.0
Qwen3-30B-Thinking	60	5.0	60	6.0	60	6.0	60	5.0
GPT-OSS-20B	20	6.0	20	5.0	20	5.0	20	6.0

A.3.2 Performance Gains Analysis

Figures 3–6 indicate that certain experts exhibit strong specialization in math-related domains. DSMoE leverages this property as a steering mechanism, encouraging MoE-based LLMs to allocate more probability mass toward domain-relevant experts.

To quantitatively validate this effect, we analyze the average expert activation frequency and router scores for the top-K specialized math experts ($K = 20$) in GPT-OSS-20B on the MMLU-Math dataset. We compare the original model (Orig.) and DSMoE under the same setting.

The results are summarized in Table 9. DSMoE significantly increases both expert selection frequency and routing scores, indicating that the model more consistently activates domain-specialized experts. This shift correlates with a substantial improvement in downstream performance, suggesting that DSMoE enhances effectiveness by strengthening expert specialization utilization rather than introducing additional capacity.

A.3.3 Small Language Models

DSMoE is effective not only for large-scale reasoning models (e.g., GPT-OSS-120B) but also for small language models (SLMs). To verify this, we evaluate DSMoE on PhiMoE-Tiny, a lightweight MoE model with 3.8B total parameters and 1.1B activated parameters.

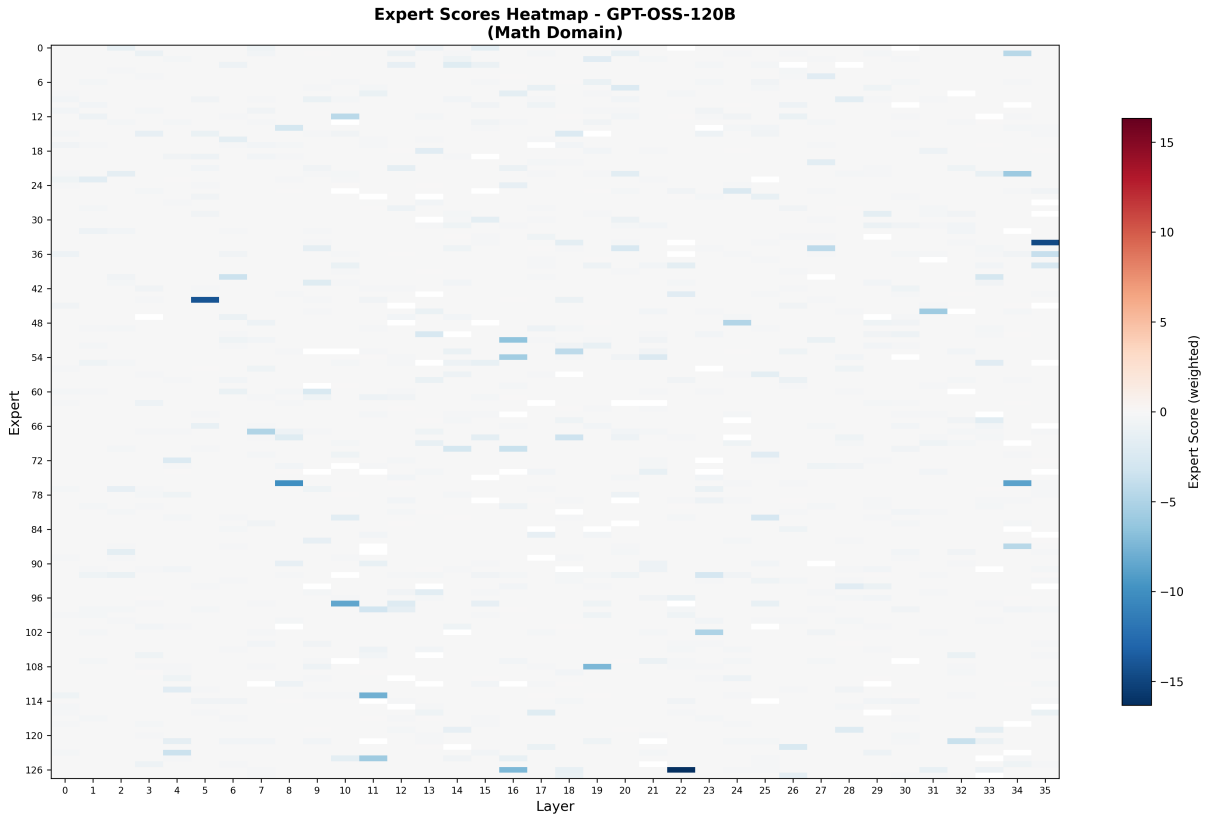


Figure 4: Domain-specific expert scores for **GPT-OSS-120B** on the Mathematics domain. Higher magnitudes indicate stronger domain specialization. Best viewed in color.

As summarized in Table 10, DSMoE consistently improves performance across all four domains (Math, Biology, Chemistry, and Physics) on the MMLU-Pro benchmark. In particular, DSMoE achieves an average gain of +6.3 points over the original baseline. These results demonstrate that DSMoE generalizes effectively across model scales and is not limited to large-capacity MoE models.

A.3.4 Domain-specific Experts versus Collaborative Experts

We analyze the roles of domain-specific experts (DE) and collaborative experts (CE) within the same layer of GPT-OSS-20B. The evaluation is conducted on 100 samples from the MMLU-Pro Math dataset.

As shown in Table 11, we identify a representative DE (Expert 30 in Layer 11) with a high average router score (0.859), indicating strong domain specialization. In contrast, a CE (Expert 10 in Layer 11) exhibits a substantially lower router score (0.136), suggesting weaker domain alignment but potential complementary contributions.

We first apply DSMoE to steer the DE and observe a +4.0 point improvement over the original model (Orig.), as shown in Table 12. To further examine the role of the CE, we ablate it by setting its router score to 0.0 while keeping all other experts unchanged. This results in a performance drop of 2.0 points compared to DSMoE.

These results suggest that while DSMoE improves performance by amplifying domain-specific experts, collaborative experts remain essential for capturing complementary knowledge. This highlights a trade-off between specialization and collaboration in MoE-based LLMs, where both types of experts jointly contribute to overall performance.

A.4 Implementation Details

Training-Free Setting. For DSMoE steering experiments, we implement our method based on the publicly available RICE implementation (Wang et al., 2025a)[1]. For *GPT-OSS-120B*, experiments are conducted on two NVIDIA H200 GPUs, while for *GPT-OSS-20B* and *Qwen3-MoE*, we utilize two NVIDIA H100 GPUs. Following RICE, our implementation leverages the vLLM library (Kwon et al., 2023), which supports parallel model loading and inference. We upgrade to vLLM version

[1] https://openreview.net/forum?id=x7fCiuCCAu

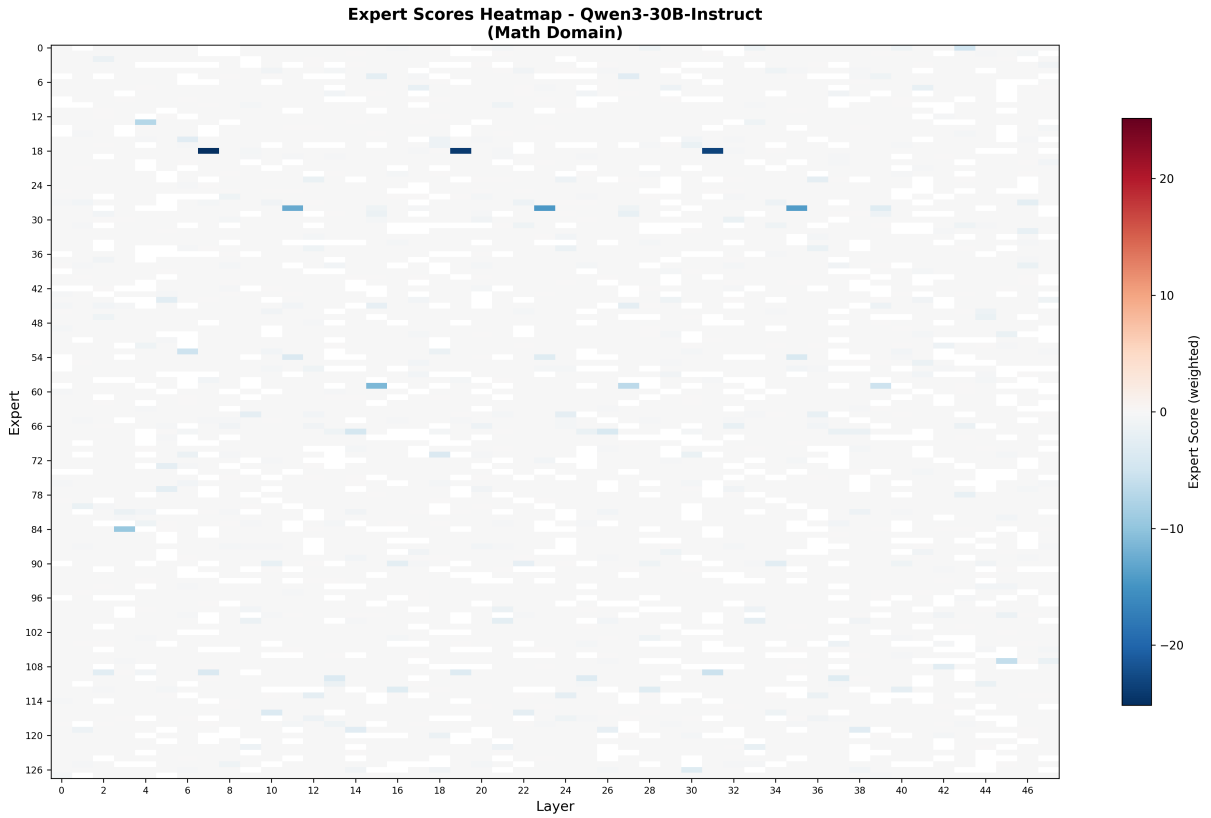


Figure 5: Domain-specific expert scores for **Qwen3-30B-Instruct** on the Mathematics domain. Higher magnitudes indicate stronger domain specialization. Best viewed in color.

0.11.0 to ensure compatibility with recent MoE-based LLMs. All models and datasets used in this work are publicly available on Hugging Face, ensuring full reproducibility of our results.

SFT Baseline. We implement the SFT baseline using the open-source PEFT library (Mangrulkar et al., 2022) with the following configuration: 3 training epochs, LoRA rank of 16, LoRA alpha of 32, and target modules including `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. This configuration results in approximately 2.7% trainable parameters relative to the total model size.

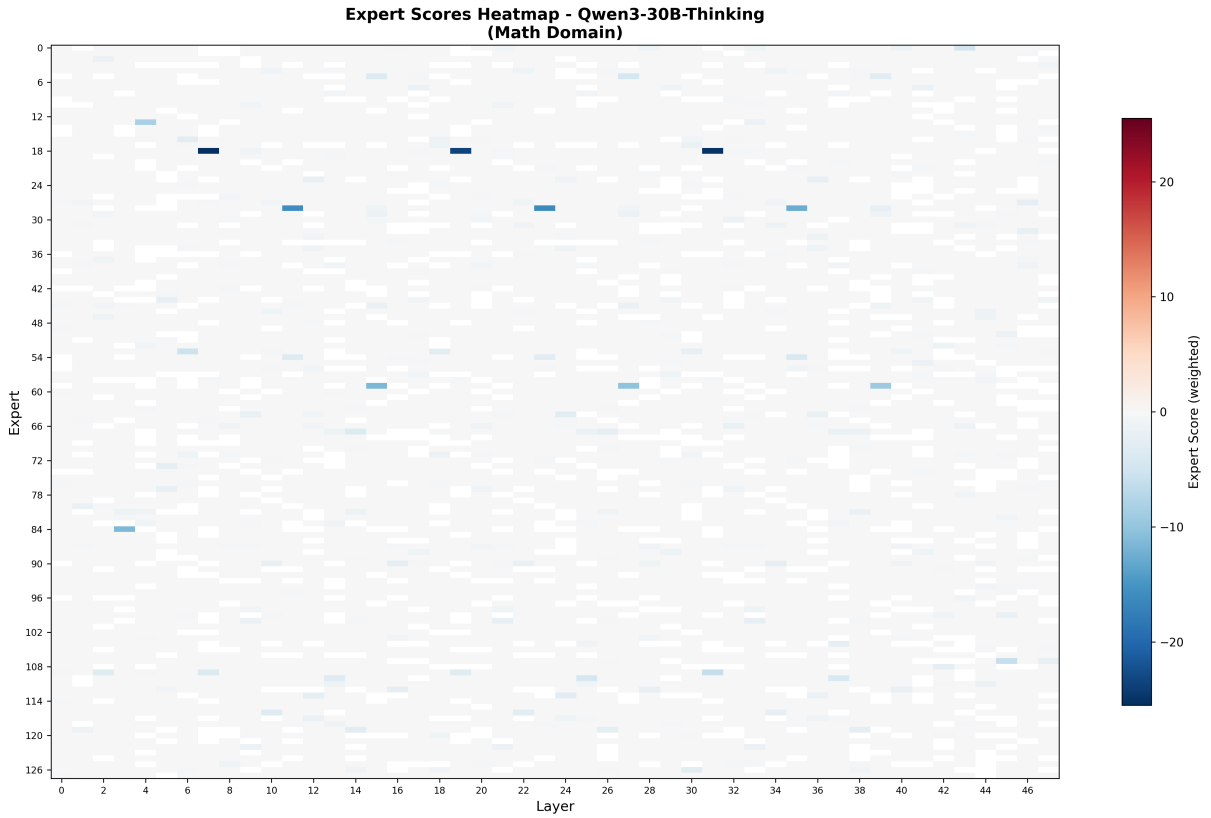


Figure 6: Domain-specific expert scores for **Qwen3-30B-Thinking** on the Mathematics domain. Higher magnitudes indicate stronger domain specialization. Best viewed in color.

Model	**Params (B)**	**Active (B)**	**Experts (N)**	**Top-K**	**Layers**	**HuggingFace Model ID**
PhiMoE-Tiny	3.8	1.1	16	2	32	`microsoft/Phi-tiny-MoE-instruct`
OLMoE	7.0	1.0	64	8	16	`allenai/OLMoE-1B-7B-0924`
Qwen1.5-MoE	14.3	2.7	60	4	24	`Qwen/Qwen1.5-MoE-A2.7B`
DeepSeek-MoE	16.0	2.8	64	6	28	`deepseek-ai/deepseek-moe-16b-base`
GPT-OSS-20B	20.0	3.6	32	4	24	`openai/gpt-oss-20b`
ERNIE-4.5	21.0	3.0	64	6	28	`baidu/ERNIE-4.5-21B-A3B-Thinking`
Qwen3MoE-Instruct	30.0	3.0	128	8	48	`Qwen/Qwen3-30B-A3B-Instruct-2507`
Qwen3MoE-Think	30.0	3.0	128	8	48	`Qwen/Qwen3-30B-A3B-Thinking-2507`
Qwen3-Next	80.0	3.0	512	10	48	`Qwen/Qwen3-Next-80B-A3B-Thinking`
GPT-OSS-120B	120.0	5.1	128	4	36	`openai/gpt-oss-120b`

Table 7: Architectural specifications of the MoE models used in our experiments. We report the Total Parameters, Activated Parameters per token, Total Number of Experts, Experts selected per token (Top-K), Number of Layers, and the corresponding HuggingFace Model ID.

Benchmark	**Domain**	**Size**	**Description**
MMLU-Pro	STEM, Law, etc.	12K	10-choice questions; minimizes random guessing
GPQA Diamond	Biology, Physics, Chemistry	198	PhD-level difficulty; 65% expert accuracy
AIME	Mathematics	30	Competition-level; integer answers (0-999)

Table 8: Summary of meta data for the evaluation benchmarks. **Size** denotes the total number of questions in the dataset (or specific subset used).

Model	Dataset	Metric	Orig.	DSMoE	Δ
GPT-OSS-20B	MMLU (Math)	Avg. Expert Frequency	0.003	0.101	**+0.098**
		Avg. Router Scores	0.161	0.433	**+0.272**
		Performance	66.5	74.5	**+8.1**

Table 9: Quantitative analysis of DSMoE on expert utilization for GPT-OSS-20B. DSMoE increases both expert activation frequency and routing confidence for domain-specialized experts, leading to improved task performance.

Dataset	Model	Total	Act.	Domain	Orig.	DSMoE	Δ
MMLU-Pro	PhiMoE-Tiny	3.8B	1.1B	Math	47.0	54.1	**+7.1**
				Biology	57.0	62.0	**+5.0**
				Chemistry	34.0	43.0	**+9.0**
				Physics	40.0	44.0	**+4.0**
				Average	44.5	50.8	**+6.3**

Table 10: Performance of DSMoE on a small language model (PhiMoE-Tiny). DSMoE consistently improves performance across all domains, demonstrating strong generalization to low-parameter regimes.

Model	Domain	Dataset	Sample	DE	DE Score	CE	CE Score
GPT-OSS-20B	Math	MMLU-Pro	100	E30–L11	0.859	E10–L11	0.136

Table 11: Identification of a domain-specific expert (DE) and a collaborative expert (CE) within the same layer. DE exhibits strong routing confidence, while CE shows lower but non-negligible activation.

Model	Domain	Dataset	Sample	Orig.	DSMoE	Δ	DSMoE w/o CE Δ
GPT-OSS-20B	Math	MMLU-Pro	100	52.0	56.0	+4.0	54.0 **-2.0**

Table 12: Ablation analysis of collaborative expert (CE). Removing CE reduces performance, highlighting its complementary role despite lower routing scores.