MIXTURE-OF-CLUSTERED-EXPERTS: Robust Management of Heterogeneous Inputs for Instruction Tuning

Anonymous ACL submission

Abstract

A sparse mixture of experts (MoE) has demonstrated its scalability by conditionally activating sub-modules without a proportional increase in FLOPs. However, standard MoE models route experts based on individual tokens, overlooking factors that can be derived from sequence-level features. In this paper, we propose a Mixture-of-Clustered-Experts (MoCE) designed to address this limitation by incorporating a dual-stage routing mechanism: the first stage performs expert group routing based on sequence-level embeddings, followed by a second stage that activates the top-k experts within the expert group at the token level. MoCE maintains the same number of activated experts as standard MoE while enabling efficient execution through k-means clustering for expert group allocation. Extensive evaluations across diverse benchmarks, including mathematical problem-solving, code generation, reasoning, and knowledge-based tasks, demonstrate that our model outperforms strong baseline performance. Notably, MoCE effectively handles heterogeneous inputs while maintaining high generalization ability. Additional in-depth analysis and case studies further validate the robustness of the proposed method. [1].

1 Introduction

A Sparse Mixture-of-Experts (MoE) has garnered significant attention for its ability to scale parameters with minimal computational overhead through the conditional activation of sub-modules (Shazeer et al., 2017; Dai et al., 2024; Jiang et al., 2024; Xue et al., 2024). In particular, the amalgamation of MoE with instruction tuning has exhibited superiority in handling complex tasks by outperforming dense models (Komatsuzaki et al., 2023; Zadouri et al., 2024; Ostapenko et al., 2023; Zhu et al., 2024).

Despite recent advancements, the gating mechanism in the MoE layer operates solely at the token level, overlooking the factors that can be derived from sequence-level attributes (Kudugunta et al., 2021a; Zhu et al., 2024). This shortcoming is particularly evident in instruction tuning scenarios, where datasets span a broad range of domains (*e.g.*, commonsense reasoning, mathematical problem-solving, code generation) and task types (*e.g.*, question answering, completion, text classification) (Longpre et al., 2023; Peng et al., 2023; Wang et al., 2022c). The MoE architecture lacks mechanisms to adequately address this semantic complexity, implying that individual experts address diverse knowledge (Dai et al., 2024). This underscores the necessity of developing a mechanism that effectively captures and manages this input heterogeneity. Furthermore, building on findings that instruction tuning functions as a regularization technique in MoE to alleviate overfitting during task-specific fine-tuning (Shen et al., 2024; Dou et al., 2024), the key challenge is to enhance the model's generalization ability by effectively managing heterogeneity while preserving these benefits.

In this work, we propose the **Mixture-of-Clustered-Experts (MoCE)**, a framework that extends the existing MoE architecture by integrating a dual-stage routing mechanism. Specifically, in the sequence-level expert group allocation, the sequence-level router predicts a cluster number based on the sequence embedding and activates an expert group mapped to the predicted cluster. We employ a k-means clustering algorithm to partition inputs, leveraging latent relationships between data points to enable more multifaceted clustering. In the subsequent token-level expert allocation process, the token-level router sparsely activates a subset of experts within the selected expert group. This hierarchical routing not only captures high-level sequence attributes but also preserves the fine-grained specificity required for effective processing.

[1] Due to anonymity, the code will be released after the review process

Moreover, MoCE preserves the number of activated experts in line with existing methods, ensuring that the efficiency of MoE is maintained.

Although MoCE manages input complexity, it inherits a key limitation of MoE architectures: loading all experts into memory can be VRAM-intensive. Therefore, we adopt a lightweight adapter-based approach, as introduced in Wu et al. (2024a), to enhance applicability for real-world practitioners.

To validate the effectiveness of MoCE, we conduct extensive evaluations across a wide range of tasks, including mathematical problem-solving, code generation, reasoning, instruction following, and knowledge-based tasks. Notably, MoCE demonstrates a prominent generalization capability, consistently outperforming baselines across most tasks, particularly exceptional results on tasks requiring specialized domain knowledge. A comprehensive case study and analysis further support the robustness of our approach. Our contributions are three-fold:

Our contributions are threefold:

- We propose the **Mixture-of-Clustered-Experts (MoCE)** framework, incorporating a dual-stage routing mechanism (at the sequence and token levels) within MoE to effectively capture the diverse objectives in the instruction tuning process.

- We maintain efficiency by preserving the number of activated experts and employing a fast-converging k-means clustering algorithm.

- Through extensive experiments across a wide range of tasks, we demonstrate the outstanding performance and generalization capabilities of MoCE, highlighting its practical applicability in real-world scenarios.

2 Related Work

Sparse Mixture-of-Experts (MoE) The Sparse Mixture of Experts (MoE) architecture, first introduced in NLP by Shazeer et al. (2017), has become a foundational framework for scaling model parameters with minimal computational cost. The architecture has been predominantly employed during the pretraining phase, serving as the backbone for a range of large language models (LLMs) (Zoph et al., 2022; Ma et al., 2018; Jiang et al., 2024; Raposo et al., 2024). However, MoE structure is highly sensitive to hyperparameter configurations (Fedus et al., 2022; Zadouri et al., 2024), leading to the development of various routing strategies, such as expert-choice and token-choice, alongside auxiliary loss functions (Du et al., 2022; Shen et al., 2024; Zoph et al., 2022). Since we focus on enabling flexible handling of heterogeneous inputs, we maintain a fixed set of hyperparameters throughout all experiments to ensure consistency.

Recently, MoE has been extended to parameter-efficient tuning techniques (Wang et al., 2022b; Diao et al., 2023). Studies have achieved exceptional tuning efficiency by applying LoRA-based parameter-efficient adaptation within the MoE framework (Zadouri et al., 2024; Wu et al., 2024b,a). Building on this direction, we construct MoCE exploiting the adapter-based upcycled model adaptation method introduced by Wu et al. (2024a).

2.1 MoE for Instruction Tuning

Studies have shown that instruction tuning within the MoE framework mitigates overfitting during task-specific fine-tuning, leading to outperforming performance and robust generalization across diverse settings (Shen et al., 2024; Komatsuzaki et al., 2023; Zadouri et al., 2024; Ostapenko et al., 2023).

Recent research has also addressed the challenge of input heterogeneity, including variations in language, domain, and task types (Zhong et al., 2022; Ren et al., 2023; Sarkar et al., 2024; Kudugunta et al., 2021b). For example, Zhao et al. (2023) introduces a two-stage routing process, combining linguistic-guided routing with dynamic expert allocation based on language difficulty. Similarly, (Gururangan et al., 2022) activates domain-specific experts during training and employs sparse activation of these experts at inference, depending on the input domain. Unlike these approaches, which apply explicit routing criteria, our method leverages clustering to capture latent input features.

3 Mixture-of-Clustered-Experts

3.1 Preliminaries

The MoE architecture replaces the dense feed-forward network (FFN) sub-layers within the Transformer block with MoE layers, each comprising a set of experts $\mathcal{E} = \{\mathcal{E}_i\}_{i=1}^{N}$. Experts are sparsely activated by a gate function $\mathcal{R}$, which routes the input token representation x in the sequence s to the most appropriate experts. The logit $h(x)$ for

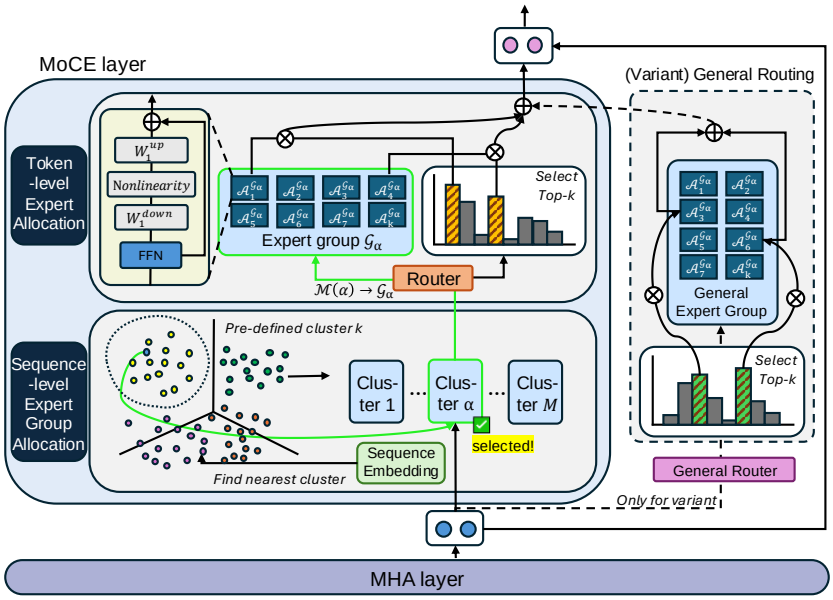


Figure 1: The overall architecture of MoCE. In the sequence-level expert group allocation, the model predicts the cluster for each input based on sequence embeddings and routes it to the corresponding expert group mapped to that cluster. For token-level expert allocation, the model calculates routing weights for each token and performs a weighted summation of the adapter outputs for the top-k selected experts. In the MoCE variant, a general router and a general expert group are additionally introduced. The output from the general expert group is summed with the results from the original MoCE.

computing the gate score is represented as:

$$h(x) = W_G^\top \cdot x, \tag{1}$$

where $W_G \in \mathbb{R}^{d_{model} \times N}$ denotes a trainable linear projection. The logit is normalized through a softmax function over the N experts, yielding the gating score $\mathcal{R}(x)_i$:

$$\mathcal{R}(x)_i = \frac{\exp^{h(x)_i}}{\sum_{j=1}^{N} \exp^{h(x)_j}}. \tag{2}$$

From the gating scores derived in this manner, only the top-k experts are conditionally selected. The routing decision and output y computation are computed as:

$$\text{SelectTopK}(r, k)_i = \begin{cases} r_i & \text{if } r_i \text{ is in the top-}k \text{ of } r, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

$$y = \sum_{i=1}^{N} \text{SelectTopK}(\mathcal{R}(x)_i, k) \cdot \mathcal{E}_i(x), \tag{4}$$

where the output from Equation 3 is considered a routing weight. The final output of the MoE layer is the weighted summation of routing weights and the activated expert outputs, as shown in Equation 4.

To address the high memory demands of MoE, we leverage the Parameter-Efficient Sparsity Crafting (PESC) method (Wu et al., 2024a). Specifically, we (1) initialize model parameters from a pre-trained dense model (Komatsuzaki et al., 2023), and (2) employ adapters to serve as experts, integrated into each FFN output while sharing the same FFN output. Therefore, the expert outputs ($\mathcal{E}_i(x)$) in Equation 4 are modified as $\mathcal{A}_i(x) = \sigma(W_i^{down} \cdot \mathcal{E}(x)) \cdot W_i^{up} + x$, where adapters $\mathcal{A}_i$ serve as lightweight expert modules.

3.2 Expert Grouping and Clustering

Our key component in MoCE is the introduction of expert groups $\{\mathcal{G}_j\}_{j=1}^{M}$, with each group containing a subset of experts $\{\mathcal{A}_i^{\mathcal{G}_j}\}_{i=1}^{N}$ and a group-specific gate function $\mathcal{R}^{\mathcal{G}_j}$.

Each expert group is then mapped to clusters derived through k-means clustering. Let $\mathcal{C}$ denote the set of clusters, and $e \in E_\alpha$ represent a sequence embedding assigned to cluster number α. The clustering objective J aims to minimize the squared L^2 distance between each embedding e and its corresponding centroid μ_α, formulated as $J = \sum_{c=1}^{|\mathcal{C}|} \sum_{e \in E_c} ||e - \mu_c||^2$. Specifically, each sequence embedding is assigned to the nearest cluster, and centroids are updated by averaging the

embeddings in each cluster as follows:

$$E_\alpha = \{e : ||e - \mu_\alpha||^2 \leq ||e - \mu_c||^2, \forall c = 1, ..., |\mathcal{C}|\},$$

$$\mu_\alpha = \frac{1}{|E_\alpha|} \sum_{e \in E_\alpha} e. \tag{5}$$

This iterative process continues until cluster assignments converge. From the clustering process, we infer that the selected expert group dynamically adapts to the specific characteristics of the input.

Rather than manually specifying the number of clusters, we adopt the elbow method, leveraging embedding values to determine the optimal cluster count. Accordingly, we identify the point where the reduction in the sum of squared distances distinctly decelerates. Although k-means clustering is often regarded as outdated, we prioritize its rapid convergence without hindering the efficiency of the MoE. As the complexity of k-means is $\mathcal{O}(|E| \cdot \mathcal{C} \cdot I \cdot D)$, where $|E|$ is the number of data points, I is the number of iterations, and D is the dimensionality of the data, its linear complexity with respect to the number of data points makes it scalable to larger datasets.

3.3 Dual-Stage Routing Mechanism

One of the compelling objectives of MoE is to "divide and conquer" by decomposing the learning task and tackling each subcomponent through adaptive expert selection, thereby promoting expert specialization (Chen et al., 2023). Based on this concept, we describe the dual-stage routing process in this section: (1) sequence-level expert group assignment and (2) token-level expert allocation. Figure 1 illustrates the overall architecture of MoCE.

Sequence-Level Expert Group Allocation We begin by applying sequence embeddings to input samples to capture latent, sequence-level features within the MoE framework. In decoder-based LLMs, representations are generated only from preceding tokens, potentially resulting in suboptimal outcomes (BehnamGhader et al., 2024). To address this, we employ an encoder-based embedding model (Wang et al., 2022a), which provides a more comprehensive contextual representation across the entire sequence.

Each sequence embedding is assigned a cluster number by the k-means model, which predicts the nearest centroid relative to the input. We then map each cluster to a corresponding expert group. Let $\mathcal{M}$ denote the mapping function that associates each cluster with its corresponding expert group:

$$\mathcal{M} : \mathcal{C} \rightarrow \mathcal{G}, \quad \mathcal{M}(\alpha) = \mathcal{G}_\alpha, \tag{6}$$

where the cluster α is mapped to expert group $\mathcal{G}_\alpha$, with $|\mathcal{C}| = |\mathcal{G}|$ and $\alpha \leq M$. Thus, all tokens in the sequence s are routed to the corresponding expert group $\mathcal{G}_\alpha$. According to this mapping function, the input sample is initially routed to the designated expert group corresponding to the assigned cluster number.

Token-Level Expert Allocation The second stage involves routing individual tokens to specific experts within the selected expert group. This token-level routing leverages the SelectTopK mechanism, identical to standard MoE layers (Equation 3). We explore two strategies: selecting the top-2 experts or applying soft merging (Zadouri et al., 2024; Muqeeth et al., 2024), where the outputs from the selected experts are combined as follows:

$$y = \sum_{i=1}^{N} \mathcal{R}^{\mathcal{G}_\alpha}(x)_i \cdot \mathcal{A}_i^{\mathcal{G}_\alpha}(x). \tag{7}$$

The outputs from each adapter are aggregated through a weighted average, with the weights generated by the gating network (Puigcerver et al., 2024). This dual-stage routing process encourages each expert group to focus on the input's specific characteristics as they process inputs within their respective clusters.

3.4 MoCE Variant

We introduce a variant of MoCE that incorporates a general expert group designed to process all input sequences. Token-level routing is performed independently within both the specialized and general expert groups, and their outputs are subsequently merged. The variant output for a token representation x routed to the α cluster group is computed as follows:

$$y = \sum_{i=1}^{N} \text{SelectTopK}\left(\mathcal{R}^{\mathcal{G}_\alpha}(x)_i, k\right) \cdot \mathcal{A}_i^{\mathcal{G}_\alpha}(x) + \sum_{j=1}^{N} \text{SelectTopK}\left(\mathcal{R}^{gen}(x)_j, k\right) \cdot \mathcal{A}_j^{gen}(x). \tag{8}$$

By combining both outputs, we aim to synthesize the knowledge acquired from the entire dataset with specialized knowledge.

Evaluation Benchmarks		# Shot	LLM-Adapter	PESC	MoCE	MoCE(+avg)	MoCE(+var)	MoCE(+avg+var)
Mathematics (EM)	GSM8K	5-Shot	21.91	27.45	30.33	**32.22**	30.63	32.15
	MATH	4-Shot	1.59	1.39	1.46	1.38	**2.19**	1.31
Code	HumanEval(Pass@1)	0-Shot	14.90	16.12	**16.45**	15.38	16.05	15.26
	HumanEval(Pass@10)	0-Shot	18.52	20.82	22.38	**22.50**	21.45	21.53
	MBPP(Pass@1)	0-Shot	16.50	20.74	20.89	20.68	**21.21**	20.76
	MBPP(Pass@10)	0-Shot	19.35	25.93	**26.07**	24.58	25.48	25.06
General (ACC)	BBH	3-Shot	36.96	37.27	38.29	38.03	37.82	**38.33**
	MMLU-Pro	5-Shot	18.99	20.34	19.95	20.39	20.39	**20.58**
	Ifeval(inst)	0-Shot	23.50	44.60	45.44	44.12	**48.08**	47.12
	Ifeval(prompt)	0-Shot	12.38	31.79	34.38	31.79	**35.67**	34.57
Reasoning & Knowledge (ACC)	GPQA	0-Shot	27.18	27.52	26.59	27.27	**29.53**	25.84
	MUSR	0-Shot	41.14	40.34	41.93	40.87	41.93	**42.59**
	Winogrande	0-Shot	66.38	67.64	**68.43**	67.48	67.56	67.40
	ARC(Easy)	0-Shot	68.90	71.38	**71.59**	**71.59**	70.75	71.09
	ARC(Challenge)	0-Shot	43.52	45.56	**46.08**	45.31	44.62	45.31
	Hellaswag	0-Shot	75.75	75.64	**75.93**	75.87	75.75	75.84
Average			31.72	35.91	36.64	36.22	**36.82**	36.55

Table 1: Performance of MoCE across four evaluation categories. We denote the MoCE with soft merging method as (+avg) and the variant incorporating general experts as (+var).

4 Experiment

4.1 Experimental Setup

Datasets. We conduct instruction tuning on MoCE using three distinct datasets. These include SlimOrca (Lian et al., 2023), a curated version of the OpenOrca dataset comprising 518K multi-task examples; Magicoder (Wei et al., 2024), which contains 110K coding problems, providing a rich source for programming and algorithmic tasks; and MetaMathQA (Zhong et al., 2022), consisting of 395K mathematical questions. In total, approximately 1M data points are used to train the models in our experiments.

Models. To mitigate the limitations of decoder-only models, we opt for E5 (Wang et al., 2022a), an encoder-based sequence embedding model. Using the elbow method, we set the optimal number of clusters to 7, with each expert group consisting of 8 experts. For instruction tuning, we train the LLaMA2 (Touvron et al., 2023) model with 7 billion parameters, selected for its wide applicability. To prevent expert collapse, we apply a load-balancing loss throughout the experiments. Detailed implementation details are provided in Appendix A.

Given that MoCE leverages adapter-based training, we compare it against two baselines: (1) LLM-Adapter, an adapter-based model designed for efficient parameter adaptation (Houlsby et al., 2019), and (2) Parameter Efficient Sparsity Crafting (PESC), which combines MoE architecture with adapters to optimize computational efficiency (Wu et al., 2024a). To ensure fairness, all models and their PESC counterparts are re-implemented with the same number of experts and equivalent hyper-parameter configurations.

Evaluation Benchmark. We evaluate the MoCE through a comprehensive benchmark. Specifically, we leverage two prominent benchmarks: Open LLM v1 (Beeching et al., 2023) and Open LLM v2 (Fourrier et al., 2024), both extensively used for assessing LLMs. Our evaluation consists of (1) **Mathematics**, where we assess the model's mathematical problem-solving capabilities using the `GSM8K` (Cobbe et al., 2021) and `MATH` (Hendrycks et al., 2021) datasets, (2) **Code**, which evaluates the model's coding proficiency and algorithmic problem-solving with the `HumanEval` (Chen et al., 2021) and `MBPP` (Austin et al., 2021) datasets, (3) **General**, where performance is tested on a range of knowledge-based NLP tasks and instruction-following abilities with datasets including `BBH` (Suzgun et al., 2022), `MMLU-Pro` (Wang et al., 2024), and `Ifeval` (instruction, prompt) (Zhou et al., 2023), and (4) **Reasoning & Knowledge**, which focuses on evaluating the model's reasoning and knowledge capacity using the `GPQA` (Rein et al., 2023), `MUSR` (Sprague et al., 2024), `Winogrande` (Sakaguchi et al., 2021), `ARC` (Easy, Challenge) (Clark et al., 2018), and `Hellaswag` (Zellers et al., 2019) datasets.

All evaluations are performed under the same settings using lm-evaluation-harness (Gao et al., 2024) and bigcode-evaluation-harness (Ben Allal et al., 2022).

Evaluation Benchmarks		# Shot	LLM-Adapter	PESC	MoCE	MoCE(+avg)	MoCE(+var)	MoCE(+avg+var)
Mathematics (EM)	GSM8K	5-Shot	29.64	56.79	57.62	**58.91**	57.32	58.68
	MATH	4-Shot	0.54	1.66	**1.96**	0.98	1.20	0.99
Code	HumanEval(Pass@1)	0-Shot	11.70	16.22	37.16	39.31	**42.41**	40.93
	HumanEval(Pass@10)	0-Shot	23.03	27.63	52.28	53.97	**54.57**	54.11
	MBPP(Pass@1)	0-Shot	17.41	19.70	40.32	40.44	40.13	**40.53**
	MBPP(Pass@10)	0-Shot	25.32	24.93	49.86	**50.48**	48.58	49.38
General (ACC)	BBH	3-Shot	39.85	47.42	46.59	**48.12**	47.91	47.21
	MMLU-Pro	5-Shot	21.97	28.44	27.76	28.27	**28.52**	24.45
	Ifeval(inst)	0-Shot	54.20	55.88	56.47	56.00	55.88	**56.71**
	Ifeval(prompt)	0-Shot	40.30	43.25	44.36	44.73	**45.66**	45.29
Reasoning & Knowledge (ACC)	GPQA	0-Shot	29.19	29.87	29.36	28.79	29.78	**30.54**
	MUSR	0-Shot	**42.20**	40.74	41.40	41.27	41.40	41.40
	Winogrande	0-Shot	73.16	73.16	73.64	**73.80**	73.48	72.85
	ARC(Easy)	0-Shot	74.54	77.86	77.53	77.06	77.31	**78.24**
	ARC(Challenge)	0-Shot	53.07	54.44	54.69	54.61	**54.86**	55.89
	Hellaswag	0-Shot	**81.18**	80.42	80.59	80.47	80.60	80.32
Average			38.58	42.40	48.22	48.58	**48.73**	48.60

Table 2: The results of applying MoCE to the Mistral family. The total number of experts used in both the baseline MoE and MoCE is the same.

5 Results and Discussion

5.1 Main Results

We report the main results of MoCE in Table 1. MoCE outperforms both baseline models, achieving an average score of up to 36.82, compared to 31.72 for LLM-Adapter and 35.91 for PESC. With the primary distinction from PESC being the integration of sequence-level routing, we find that the observed performance gains can be attributed to effectively routing heterogeneous inputs to the appropriate expert groups based on features extracted at the sequence level.

In our approach, inputs with similar sequence-level features are routed to the same expert group, which could potentially increase the risk of overfitting within these groups. To address this, our objective is to manage diverse inputs flexibly while retaining the regularization benefits of instruction tuning. As a result, MoCE achieves higher accuracy in 14 out of 16 evaluation tasks compared to PESC, particularly in zero-shot and few-shot settings, demonstrating strong generalization across a broad range of tasks. These findings suggest that MoCE successfully preserves the regularization effects of instruction tuning without introducing performance degradation due to overfitting.

5.2 Application of MoCE Across Different Model Families

To examine the adaptability of MoCE across different model families, we extend the framework to the Mistral family (Mistral 7B). As shown in Table 2, MoCE continues to deliver robust performance, with remarkable gains on domain-specific benchmarks such as HumanEval (from 16.22 to 42.41) and MBPP (from 19.70 to 40.53). These promising results underscore the effectiveness of MoCE in handling specialized tasks, where its structured routing mechanism allows for better expert training. Furthermore, the model maintains competitive performance across other benchmarks such as mathematics, reasoning, and general knowledge tasks, achieving either on-par or better results compared to the baseline. These results demonstrate that MoCE provides a robust solution across various model families beyond LLaMA.

Method	Math	Code	General	R&K
MoCE (E5)	15.90	21.45	34.52	55.09
LLM+Adapter	11.75	17.32	22.96	53.81
PESC (E-40)	14.89	20.53	34.59	54.57
Instructor Embedding				
MoCE	16.71	**21.34**	34.96	55.07
MoCE(+avg)	**17.69**	20.84	**35.15**	**55.18**
MoCE(+var)	17.10	21.07	34.65	54.94
MoCE(+var+avg)	17.06	21.33	34.67	54.51

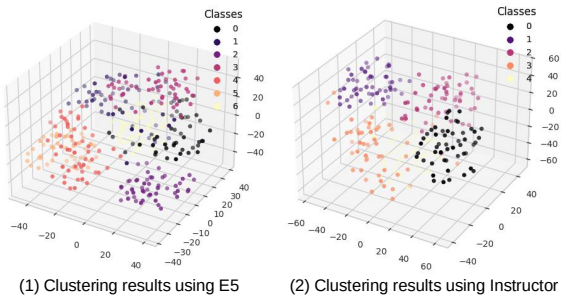
Table 3: Results of leveraging the Instructor embedding method for clustering. We present the average score across each evaluation category. PESC is trained with 40 experts, aligning with the Instructor embedding setup (5 expert groups, 8 experts each). Detailed results are provided in Appendix B.1

5.3 Impact of Modifying MoCE Components

To gain deeper insights into MoCE, we investigate its efficacy with different embedding and clustering

Method	Math	Code	General	R&K
MoCE (E5)	15.90	21.45	34.52	55.09
LLM+Adapter	11.75	17.32	22.96	53.81
PESC (E-32)	15.10	21.20	33.73	**54.92**
Gaussian Mixture Model (GMM)				
MoCE	15.61	20.97	33.83	54.69
MoCE(+avg)	16.07	21.23	34.03	54.66
MoCE(+var)	15.86	20.92	**34.79**	54.75
MoCE(+var+avg)	**16.50**	**22.05**	34.21	54.57

Table 4: Results of leveraging Gaussian mixture model for clustering. We present the average score across each evaluation category. PESC is trained with 32 experts, aligning with the GMM setup (4 expert groups, 8 experts each). Detailed results are further provided in Appendix B.2.



(1) Clustering results using E5 (2) Clustering results using Instructor

Figure 2: K-means clustering results using E5 and Instructor embedding method

methods beyond the original approach.

Initially, we replace the text embedding model with Instructor (Su et al., 2023), a model designed to generate text embeddings based on input and instructions. Applying the elbow method, we find the optimal number of clusters to be 5, which is fewer than the 7 clusters used in the E5 model. As shown in Table 3, MoCE with the Instructor embedding consistently outperforms PESC (E-40), a counterpart model configured with the same number of experts across all evaluation categories. Notably, the model exhibits competitive performance even when compared to E5-based MoCE. These findings suggest that integrating advanced embedding methods capable of better capturing the latent characteristics of inputs could enhance its ability to manage heterogeneous inputs more comprehensively.

Our next focus shifts to the clustering approach. While we leverage k-means clustering for its simplicity and fast convergence, we also evaluate the impact of applying the Gaussian Mixture Model (GMM) clustering algorithm, which considers the direction of data distribution. As shown in Table 4, while MoCE with the GMM algorithm achieves comparable performance gains relative to PESC, no improvements are observed in reasoning and knowledge tasks. We conjecture that the model may have learned overly simplistic patterns from the input data, resulting in diminished generalization. Since GMM introduced delays due to its computational complexity, we conclude that the k-means clustering algorithm provides a more effective approach for our framework, offering a better balance between efficiency and performance. Further details on the results regarding the modification of the embedding and clustering method are provided in Appendix B.

5.4 Impact of Expert Averaging and General Expert Aggregation in MoCE Variant

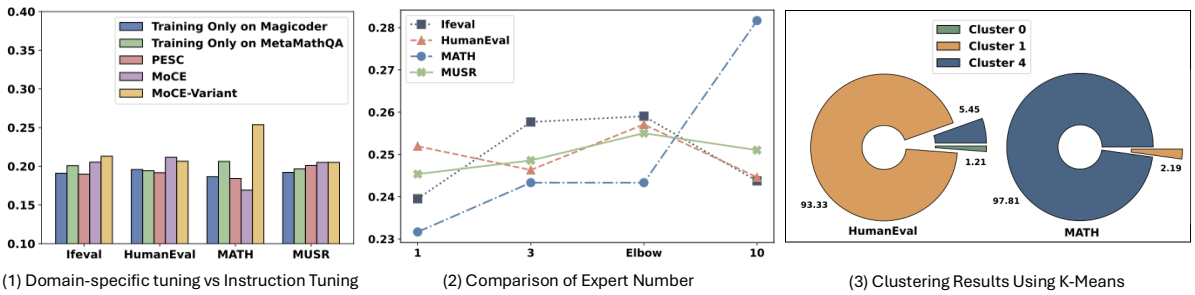
In Table 1 and Table 2, the variant approach (MoCE(+var) exhibits the highest scores, whereas in Table 3 and Table 4, the soft merging method demonstrates greater effectiveness. The results from the former indicate that the variant model improves performance through a more collaborative approach, effectively integrating general experts with specialized experts. Furthermore, the findings from the latter underscore the advantage of the soft merging method, which leverages exact gradient calculation during model training rather than gradient estimation (Muqeeth et al., 2024). However, we observe such methods do not always guarantee consistently superior performance.

5.5 Embedding Results

We present a visualization of the clustering results for 50 data points per cluster, utilizing both E5 and Instructor embeddings. Dimensionality reduction is performed via t-SNE, and the resulting plots are depicted in Figure 2. Both methods exhibit strong convergence around their respective centroids. These embedding results play a crucial role in enhancing the generalization performance of the MoCE model across various tasks. The clear separation of clusters demonstrates the model's potential to more effectively handle the diversity of input data.

5.6 How Effectively Does MoCE Manage Inputs and Benefit from Instruction Tuning?

In this section, we aim to investigate how effectively MoCE handles heterogeneous inputs while benefiting from instruction tuning. For the compar-



(1) Domain-specific tuning vs Instruction Tuning (2) Comparison of Expert Number (3) Clustering Results Using K-Means

Figure 3: (1) Comparison results of domain-specific tuning and instruction tuning-based methods, (2) Performance fluctuations based on varying the number of clusters. We report results from one benchmark in each evaluation category, (3) Distribution of routed cluster numbers for the evaluation benchmark. Results are reported for two distinct domain-specific benchmarks.

ison model, we train the baseline PESC on domain-specific datasets (mathematics and code, respectively) and compare its performance with MoCE.

As plotted in Figure 3 (1), MoCE and its variant models consistently outperform models trained solely on individual domains (Training Only on Magicoder, Training Only on MetaMathQA), even surpassing their performance in the specific domains of code and math. We conjecture that effectively managing diverse inputs fosters expert specialization while enhancing generalization across tasks with differing requirements. We conclude that, rather than training on tasks individually, handling them comprehensively while allowing each expert group to specialize is a more effective approach, leading to performance gain and stronger generalization capabilities.

5.7 How Should the Number of Experts Be Determined?

We employ the elbow method to identify the optimal number of clusters for MoCE and further investigate performance variations under different cluster configurations. Figure 3 (2) illustrates the performance trends as the number of clusters changes. Setting the number of clusters to 7, as suggested by the elbow method, delivers the best overall performance. This demonstrates the effectiveness of the elbow method in determining the optimal number of experts. The results align with previous research, which highlights that increasing the number of experts does not always lead to proportional improvements in model performance (Zadouri et al., 2024; Chen et al., 2023), emphasizing the importance of carefully setting the number of experts.

5.8 How Are Domain Inputs Clustered?

We analyze the distribution of expert group assignments across two distinct domain-specific evaluation benchmarks—HumanEval and MATH. Figure 3 (3) shows that MATH is predominantly routed to Cluster 4, while HumanEval is mainly assigned to Cluster 1. This observation suggests that inputs from different domains are allocated to distinct expert groups, leading to performance improvements, as shown in Figure 3 (1). Namely, the expert group routing results facilitate expert specialization by assigning specific tasks to dedicated expert groups. However, it is essential to recognize that the criteria for data partitioning are not solely domain-dependent, resulting in diverse groupings beyond the domain.

6 Conclusion

In this work, we addressed the limitation of the MoE model's token-based routing, which fails to capture important sequence-level features required for managing complex and heterogeneous inputs. To tackle this issue, we proposed the Mixture-of-Clustered-Experts (MoCE) framework, featuring a novel dual-stage routing mechanism. By incorporating a sequence-level expert group allocation and token-level expert allocation, MoCE effectively manages inputs while maintaining computational efficiency. Our experimental results consistently show that MoCE outperforms baseline models, highlighting its ability to foster expert specialization and maintain robust generalization. Comprehensive evaluations further validated the effectiveness of our approach, establishing MoCE as a promising framework for handling diverse and complex inputs.

Limitations

While MoCE introduces a novel dual-stage routing mechanism that improves model performance across various tasks, several limitations remain. First, due to computational resource constraints, we applied MoCE to adapter-based models to maintain efficiency. Applying this method directly to the full feed-forward network (FFN) layers for instruction tuning is left as future work.

Second, interpretability is an ongoing challenge in MoE-based architectures. Although we employed k-means clustering for its efficiency, this method still does not address the explainability issues inherent in MoE systems.

Lastly, while MoCE has demonstrated strong results across tasks in mathematics, coding, and general knowledge, expanding the framework to more languages and specialized domains remains an open area for future work. Investigating its effectiveness in multilingual and domain-specific settings will be essential for broadening the applicability of MoCE.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard (2023-2024). https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

Loubna Ben Allal, Niklas Muennighoff, Logesh Kumar Umapathi, Ben Lipkin, and Leandro von Werra. 2022. A framework for the evaluation of code generation models. https://github.com/bigcode-project/bigcode-evaluation-harness.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Tianlong Chen, Zhenyu Zhang, AJAY KUMAR JAISWAL, Shiwei Liu, and Zhangyang Wang. 2023. Sparse moe as the new dropout: Scaling dense and self-slimmable transformers. In *The Eleventh International Conference on Learning Representations*.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297, Bangkok, Thailand. Association for Computational Linguistics.

Shizhe Diao, Tianyang Xu, Ruijia Xu, Jiawei Wang, and Tong Zhang. 2023. Mixture-of-domain-adapters: Decoupling and injecting domain knowledge to pretrained language models' memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5113–5129, Toronto, Canada. Association for Computational Linguistics.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. 2024. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, 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.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2022. DEMix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, Seattle, United States. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *Preprint*, arXiv:2103.03874.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

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, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. 2023. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *The Eleventh International Conference on Learning Representations*.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. 2021a. Beyond distillation: Task-level mixture-of-experts for efficient inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3577–3599, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. 2021b. Beyond distillation: Task-level mixture-of-experts for efficient inference. *arXiv preprint arXiv:2110.03742*.

Wing Lian, Guan Wang, Bleys Goodson, Eugene Pentland, Austin Cook, and Chanvichet Vong. 2023. Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification. *Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification*, 5.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1930–1939.

Mohammed Muqeeth, Haokun Liu, and Colin Raffel. 2024. Soft merging of experts with adaptive routing.

Oleksiy Ostapenko, Lucas Caccia, Zhan Su, Nicolas Le Roux, Laurent Charlin, and Alessandro Sordoni. 2023. A case study of instruction tuning with mixture of parameter-efficient experts. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, and Neil Houlsby. 2024. From sparse to soft mixtures of experts. In *The Twelfth International Conference on Learning Representations*.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. 2024. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*.

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.

Xiaozhe Ren, Pingyi Zhou, Xinfan Meng, Xinjing Huang, Yadao Wang, Weichao Wang, Pengfei Li, Xiaoda Zhang, Alexander Podolskiy, Grigory Arshinov, et al. 2023. Pangu-{\Sigma}: Towards trillion parameter language model with sparse heterogeneous computing. *arXiv preprint arXiv:2303.10845*.

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

Soumajyoti Sarkar, Leonard Lausen, Volkan Cevher, Sheng Zha, Thomas Brox, and George Karypis. 2024. Revisiting smoe language models by evaluating inefficiencies with task specific expert pruning. *arXiv preprint arXiv:2409.01483*.

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. In *International Conference on Learning Representations*.

Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Y Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou. 2024. Mixture-of-experts meets instruction tuning: A winning combination for large language models. In *The Twelfth International Conference on Learning Representations*.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *Preprint*, arXiv:2310.16049.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *Preprint*, arXiv:2210.09261.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022a. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022b. AdaMix: Mixture-of-adaptations for parameter-efficient model tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5744–5760, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022c. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2024. Magicoder: Empowering code generation with oss-instruct. In *Forty-first International Conference on Machine Learning*.

Haoyuan Wu, Haisheng Zheng, and Bei Yu. 2024a. Parameter-efficient sparsity crafting from dense to mixture-of-experts for instruction tuning on general tasks. *arXiv preprint arXiv:2401.02731*.

Xun Wu, Shaohan Huang, and Furu Wei. 2024b. Mixture of loRA experts. In *The Twelfth International Conference on Learning Representations*.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. 2024. Openmoe: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2402.01739*.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. 2024. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *The Twelfth International Conference on Learning Representations*.

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

Xinyu Zhao, Xuxi Chen, Yu Cheng, and Tianlong Chen. 2023. Sparse moe with language guided routing for multilingual machine translation. In *The Twelfth International Conference on Learning Representations*.

Tao Zhong, Zhixiang Chi, Li Gu, Yang Wang, Yuanhao Yu, and Jin Tang. 2022. Meta-dmoe: Adapting to domain shift by meta-distillation from mixture-of-experts. *Advances in Neural Information Processing Systems*, 35:22243–22257.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and

Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

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*.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*.

A Implementation Details

The hyperparameters used in our experiments are as follows: the adapter dimension is set to 64, with an MoE scaling factor of 0.25, and the number of experts set to 8. The maximum sequence length is 512, and we apply either top-2 selection or soft merging for expert routing. The learning rate is set to 2×10^{-4}, and a batch size of 16 is used. All models are trained for 1 epoch on 8 A100 GPUs.

Two embedding models are used for clustering: E5 (`intfloat/e5-large-v2`) and Instructor (`hkunlp/instructor-xl`), both initialized from Huggingface checkpoints. Similarly, two models are used for instruction tuning: LLaMA2 (`meta-llama/Llama-2-7b-chat-hf`) and Mistral (`mistralai/Mistral-7B-Instruct-v0`), also initialized from Huggingface checkpoints.

B Case Studies

B.1 Results Using Instructor Embedding

Table 5 presents the results of clustering based on the Instructor embedding model instead of E5. Instructor embeddings generate text embeddings based on both instructions and inputs, offering a versatile approach capable of handling a wide range of tasks and domains. We observe that the modified embedding method is also effective, suggesting that applying alternative embedding techniques that capture latent sequence-level information more effectively could lead to further performance improvements.

B.2 Results Using the GMM Algorithm

Table 6 provides the results of applying the Gaussian Mixture Model (GMM) clustering algorithm, which accounts for the directional distribution of data, as an alternative to the k-means clustering algorithm. Although GMM theoretically provides better handling of data variance, the empirical results show marginal differences in performance compared to k-means clustering.

C Data Samples Belonging to Each Cluster

We select one sample from each cluster and conduct a qualitative analysis. The analysis results for the data in each cluster are as follows. We observe that the content within each cluster appears mostly distinct.

Cluster 1: Premise: "A man and woman walk away from the camera as a redheaded man walks in the other direction."Hypothesis: "A red haired man walking towards the camera."Do we know that the hypothesis entailed by the premise? Step by step answer:

Cluster 2: "Definition: In this task, you're given the title and three arbitrary sentences out of a five-sentence story. You are also given three additional sentence options, a, b, and c, that may or may not belong to the story. Your job is to pick the two options that seamlessly connect with the rest of the story; note that the selected choices may fit into the story at any point. Indicate your selected choices in order, i.e., as 'ab'/'bc'/'ac' and not as 'ba'/'ca'/'cb'. If options are equally plausible, pick the ones that make more sense.Input: Title: Toni was getting some groceries. When she was about to check out, she was stopped. Toni was very upset. Choices: a. She ended up throwing something at the woman. b. Another woman told Toni she was there first. c. Everyone around the room looked at each other.Output:"

Cluster 3: Answer the following question: Title: BOOOOOOO ! ! ! Review: Come on people it seems like every time Toby Mac makes a remix cd it's just a version of his songs with no guitar and lots of piano ? The only reason I even gave this two stars is because well it's still Toby and no matter how hard he tries he'll still be okay :) Don't get me wrong I looooooove his music just not his mixes . Is this product review negative?Choices: a). Yes; b). No;Answer:

Cluster 4: How would someone go about accomplishing this goal?what to use in place of a cocktail shaker

Cluster 5: After saving $25 in his piggy bank every month for 2 years, John had to use $400 from his savings to fix his car. How much money is left in his piggy bank?

Cluster 6: Randy, Peter, and Quincy all drew pictures. Peter drew x pictures. Quincy drew 20 more pictures than Peter. If they drew 41 pictures altogether, Randy drew 5. What is the value of unknown variable x? What is the value of unknown variable x?

Cluster 7: QUESTION: Premise: "A radio announcer in a gray sport jacket airing his show from tech pollo summit."Based on this premise, can we conclude that the hypothesis "The announcer is wearing gray." is true?Options:- yes- it is not possible to tell- noLetś solve it slowly: A announcer wore gray when his show aired from Tech Pollo Summit.The answer is yes.[QUESTION] Given the sentence "A monk shooting a bow and arrow at the target." can we conclude that "A target getting shot at."?Options:- yes- it is not possible to tell- noA target getting shot at is directly implied from shooting a bow and arrow at the target.The answer is yes.Q: Premise: "A pretty girl dressed in canadian colors and with a painted canadian flag on her cheek eats some food in a bowl."Hypothesis: "A girl is petting a stray dog at the park."Is the hypothesis entailed by the premise?Options:- yes- it is not possible to tell- noA: A girl that eats cannot be petting a dog at the same time.The answer is no.QUESTION: Can we conclude from "A man works on a large vehicle." that "Mechanic fixes the problem."?Options:- yes- no- it is not possible to tellLetś solve it slowly: A man can work on a large vehicle without being a mechanic.The answer is it is not possible to tell.[QUESTION] If "Three women sitting on a dock splashing their feet in the water." does that mean that "Three woman sit with their feet in the water."?Options:- yes- it is not possible to tell- noMost people sitting with their feet in the water splash water with them.The answer is yes.Q: Given the sentence "A man in a blue shirt is walking in front of a store called celtic ireland." can we conclude that "A man is in public."?Options:- yes- it is not possible to tell- noA:

Evaluation Benchmarks		# Shot	LLM+Adapter	PESC	MoCE	MoCE(+avg)	MoCE(+var)	MoCE(+var+avg)
Mathematics (EM)	GSM8K	5-Shot	21.91	28.13	31.99	**33.36**	32.37	32.37
	MATH	4-Shot	1.59	1.65	1.43	**2.01**	1.83	1.74
Code	HumanEval(Pass@1)	0-Shot	14.90	16.62	16.29	15.91	**16.69**	16.41
	HumanEval(Pass@10)	0-Shot	18.52	20.69	21.56	21.80	21.79	**22.89**
	MBPP(Pass@1)	0-Shot	16.50	20.20	**21.57**	20.92	20.68	21.25
	MBPP(Pass@10)	0-Shot	19.35	24.60	**25.95**	24.73	25.13	24.78
General (ACC)	BBH	3-Shot	36.96	37.48	37.79	38.03	37.32	**38.27**
	MMLU-Pro	5-Shot	18.99	20.41	**20.84**	20.68	20.70	20.37
	Ifeval(inst)	0-Shot	23.50	46.64	46.64	**47.12**	46.76	46.40
	Ifeval(prompt)	0-Shot	12.38	33.83	34.57	**34.75**	33.83	33.64
Reasoning & Knowledge (ACC)	GPQA	0-Shot	27.18	27.52	28.86	29.28	**29.45**	26.85
	MUSR	0-Shot	41.14	40.21	39.95	40.74	**41.27**	40.74
	Winogrande	0-Shot	66.38	67.01	67.48	67.25	**67.72**	67.01
	ARC(Easy)	0-Shot	68.90	71.59	**71.97**	71.63	70.96	71.09
	ARC(Challenge)	0-Shot	43.52	45.48	46.33	**46.42**	44.80	45.56
	Hellaswag	0-Shot	75.75	75.60	**75.85**	75.75	75.41	75.81
Average			33.54	38.03	38.68	**38.85**	38.56	38.39

Table 5: Entire results of leveraging the Instructor embedding method for clustering

Evaluation Benchmarks		# Shot	LLM+Adapter	PESC	MoCE	MoCE(+avg)	MoCE(+var)	MoCE(+var+avg)
Mathematics (EM)	GSM8K	5-Shot	21.91	29.04	**29.72**	**30.55**	29.80	31.61
	MATH	4-Shot	1.59	1.15	1.49	**1.59**	**1.92**	1.38
Code	HumanEval(Pass@1)	0-Shot	14.90	16.19	15.86	15.69	**16.08**	**16.92**
	HumanEval(Pass@10)	0-Shot	18.52	21.10	21.80	23.28	**23.59**	**24.93**
	MBPP(Pass@1)	0-Shot	**16.50**	**21.68**	20.64	20.93	19.81	20.77
	MBPP(Pass@10)	0-Shot	**19.35**	**25.82**	25.56	25.03	24.20	25.56
General (ACC)	BBH	3-Shot	36.96	37.79	37.09	37.25	**37.88**	**37.98**
	MMLU-Pro	5-Shot	18.99	20.73	**20.06**	**20.81**	20.43	20.69
	Ifeval(inst)	0-Shot	23.50	44.24	45.44	**45.32**	**46.28**	45.08
	Ifeval(prompt)	0-Shot	12.38	32.16	32.72	**32.72**	**34.57**	33.09
Reasoning & Knowledge (ACC)	GPQA	0-Shot	27.18	27.01	**27.27**	**27.43**	26.93	26.17
	MUSR	0-Shot	41.14	41.80	40.61	**41.14**	**42.46**	41.53
	Winogrande	0-Shot	**66.38**	**67.96**	**67.96**	67.88	66.77	67.09
	ARC(Easy)	0-Shot	**68.90**	**71.80**	71.30	71.13	71.04	71.21
	ARC(Challenge)	0-Shot	43.52	45.22	45.05	44.62	**45.65**	**45.73**
	Hellaswag	0-Shot	75.75	**75.70**	**75.93**	75.77	75.63	75.71
Average			33.54	38.03	37.94	**38.06**	**38.23**	38.21

Table 6: Entire results of leveraging the Gaussian mixture model for clustering