Sparse Mixture of Experts as Unified Competitive Learning

Anonymous ACL submission

Abstract

Sparse Mixture of Experts (SMoE) improves the efficiency of large language model training by directing input tokens to a subset of experts. Despite its success in generation tasks, its generalization ability remains an open question. In this paper, we demonstrate that current SMoEs, which fall into two categories: (1) Token Choice ;and (2) Expert Choice, struggle with tasks such as the Massive Text Embedding Benchmark (MTEB). By analyzing their mechanism through the lens of competitive learning, our study finds that the Token Choice approach may overly focus on irrelevant experts, while the Expert Choice approach risks discarding important tokens, potentially affecting performance. Motivated by this analysis, we propose Unified Competitive Learning SMoE (USMoE), a novel and efficient framework designed to improve the performance of existing SMoEs in both scenarios: with and without training. Extensive experiments across various tasks show that USMoE achieves up to a **10%** improvement over traditional approaches or reduces computational inference costs by **14%** while maintaining strong performance.

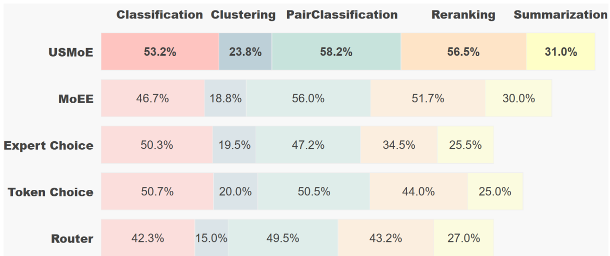


Figure 1: We compare the performance of USMoE (ours) with the Expert Choice and Token Choice approaches on the Massive Text Embedding Benchmark (MTEB). The results demonstrate that our method outperforms traditional approaches across six tasks using OLMoE-7B (Muennighoff et al., 2024) without additional training.

1 Introduction

Sparse Mixture of Experts (SMoE) models have achieved notable success in natural language processing (NLP) and visual representation learning tasks (Du et al., 2022; Fedus et al., 2022; Riquelme et al., 2021a; Shen et al., 2023). These advancements build on the Transformer architecture (Vaswani et al., 2017) and its variants (Child et al., 2019; Dai et al., 2019b), which leverage large datasets and significant compute resources. However, training large Transformer models can be prohibitively expensive, requiring extensive compute hours (Kaddour et al., 2023). To overcome this issue, SMoE models activate only a subset of experts for each input, reducing inference time compared to dense models (Shazeer et al., 2017a; Zoph et al., 2022; Artetxe et al., 2022; Krajewski et al., 2024). The SMoE architecture can be categorized into two variants: *Token Choice*, which assigns experts to each token (Dai et al., 2024; Team, 2024; Muennighoff et al., 2024; Jiang et al., 2024a), and *Expert Choice*, which assigns tokens to each expert (Zhou et al., 2022b). The advantage of Token Choice lies in its ability to dynamically select experts for each token, while Expert Choice ensures a more balanced token distribution across experts.

Despite their promising results, SMoE models have several limitations. The Expert Choice approach suffers from token dropping (Zhou et al., 2022b), while the Token Choice approach struggles with unbalanced expert loading (Shazeer et al., 2017b). Additionally, both approaches are prone to representation collapse, where either a few experts dominate the routing or all experts learn similar representations (Chi et al., 2022a; Chen et al., 2022). Recent research has explored improving router policies (Chi et al., 2022b; Chen et al., 2023a; Do et al., 2023a) to mitigate these issues. However, existing methods face two key challenges: (1) the use of auxiliary losses requires balancing router loss and task loss, leading to trade-offs, and (2) they still struggle with the fundamental limitations of either the Token Choice or Expert Choice approach.

*Which approach is better for SMoE - **Token Choice** or **Expert Choice** - in terms of generalization across multiple tasks, both with and without training?*

In this paper, we address this question by reexamining SMoE through the lens of Competitive Learning (Rumelhart and Zipser, 1985; Kaski and Kohonen, 1994; Srivastava et al., 2013; Pham et al., 2024a). From this perspective, Token Choice can be seen as horizontal competitive learning, where the most similar expert is selected for each token, while Expert Choice represents vertical competitive learning, where each expert selects the most similar tokens. This viewpoint reveals a key trade-off: Expert Choice risks dropping important tokens, whereas Token Choice must process both relevant and irrelevant tokens.

Building on this analysis, we propose **Unified Competitive Learning SMoE (USMoE)**, a robust and efficient framework comprising two key components: (1) **Unified Competitive Score** and (2) **Unified Competitive Mechanism**. These components enable the SMoE model to dynamically prioritize tokens or experts while ensuring the selection of the most similar token-expert pair, enhancing both robustness and effectiveness. To demonstrate the effectiveness of our approach, we evaluate USMoE across multiple scenarios, including pretraining and both fine-tuned and non-fine-tuned settings. USMoE consistently outperforms baseline methods across these scenarios, with particularly strong gains in tasks that require deep input understanding, such as semantic textual similarity, classification, and clustering. Extensive experiments across various benchmarks show that USMoE achieves up to a **10%** improvement over traditional approaches or reduces inference computational costs by **14%**, all while maintaining high performance.

In summary, this paper makes the following key contributions:

- We introduce a **Competitive Learning** perspective on SMoE, highlighting the weaknesses of existing approaches.
- We propose **USMoE**, a robust and efficient framework that addresses the limitations of both Token Choice and Expert Choice.
- We **theoretically demonstrate** that USMoE effectively mitigates representation collapse, outperforming baseline methods.
- We conduct **extensive experiments** on large language models, covering pretraining and both fine-tuned and non-fine-tuned settings, providing a detailed analysis of USMoE's performance and effectiveness.

2 Related Work

Sparse Mixture of Experts (SMoE). Sparse Mixture of Experts (SMoE), an extension of the Mixture of Experts framework (Jacobs et al., 1991; Jordan and Jacobs, 1994), has gained traction with large language models and has since been applied in various domains, including computer vision and speech recognition (Zhou et al., 2022c; Riquelme et al., 2021b). The SMoE architecture consists of two main variants: **Token Choice**, where experts are assigned to each token (Shazeer et al., 2017b; Fedus et al., 2022; Jiang et al., 2024b; Do et al., 2024b), and **Expert Choice**, where tokens are assigned to specific experts (Zhou et al., 2022b).

Token Choice treats all tokens equally, which has raised concerns among researchers (Wu et al., 2021; Hou et al., 2022; Lin et al., 2025), while Expert Choice suffers from token-dropping issues. Additionally, SMoE faces the challenge of representation collapse, where experts produce similar outputs. Various solutions have been proposed, such as XMoE, which employs low-dimensional routing scores (Chi et al., 2022b), and SMoE-dropout, which gradually activates more experts (Chen et al., 2023a). Other approaches, including HyperRouter (Do et al., 2023a) and StableMoE (Dai et al., 2022a), focus on enhancing router stability and robustness. Although these advancements have improved SMoE models, representation collapse remains a persistent issue (Pham et al., 2024a; Do et al., 2024a). Our approach addresses this by optimizing the alignment between tokens and the most suitable experts, expanding expert specialization and mitigating collapse.

Competitive Learning. Competitive learning is an *unsupervised learning* approach where computational units compete to respond to a given input, enabling feature discovery (McClelland and Rumelhart, 1987; Andersen et al., 1969; Stefanis and Jasper, 1969). Inspired by the biological brain, this concept has recently been leveraged to enhance the efficiency of Large Language Models (LLMs) (Cai et al., 2024; Zhao et al., 2024). Furthermore, competitive learning has been shown to enhance the efficiency of *Mixture of Experts (MoE)* models (Ahn and Sentis, 2021; Cai et al., 2023).

The most relevant work to our work is **CompeteSMoE** (Pham et al., 2024b), which frames competition through expert outputs and employs a router trained to predict competition outcomes in a scheduled manner. However, it shares a key limitation with traditional competitive learning frameworks, activating all experts, which becomes infeasible for large-scale models with millions of parameters. In contrast, we redefine competition among expert embeddings into two categories: **Token Choice SMoE**, which represents *horizontal competitive learning*, and **Expert Choice SMoE**, which represents *vertical competitive learning*. Building on this perspective, we introduce two novel scoring methods and mechanisms - Unified Competitive Score and Unified Competitive Mechanism - that enhance traditional SMoE in both performance and efficiency.

3 Methodology

We introduce Unified Competitive Learning SMoE (USMoE), a novel and efficient Sparse Mixture of Experts framework designed to address the limitations of both Token Choice and Expert Choice through a unified competitive learning mechanism. As shown in Section 3.2, our approach consists of two key components:

- Unified Competitive Score – a scoring function that balances expert selection.
- Unified Competitive Mechanism – a structured routing strategy that ensures efficient and effective expert allocation.

3.1 Preliminaries

Sparse Mixture of Experts. The Sparse Mixture of Experts (SMoE) is typically a transformer architecture that replaces the multi-layer perceptron (MLP) layers in standard transformers with Mixture of Experts (MoE) layers, inspired by (Shazeer et al., 2017a). Given $\mathbf{x} \in \mathbb{R}^{b \times d}$ as the output of the multi-head attention (MHA) layer, the result of the SMoE with n experts is a weighted sum of each expert's computation, $E_i(x)$, weighted by the router function $\mathcal{S}(x)$:

$$f_{\text{SMoE}}(\boldsymbol{x}) = \sum_{i=1}^{n} \mathcal{S}(\boldsymbol{x})_i \cdot E_i(\boldsymbol{x}) \qquad (1)$$

Where $\mathcal{S}(x)$ is computed by $TopK$ function as below the Equation 2, and W_e is a learnable experts embeddings.

$$\mathcal{S}(\boldsymbol{x}) = \text{TopK}(\text{softmax}(W_e \cdot x), k) \qquad (2)$$

$$\text{TopK}(\boldsymbol{v}, k) = \begin{cases} \boldsymbol{v_i} & \text{if } \boldsymbol{v_i} \text{ is in the top } k \text{ largest of } \boldsymbol{v}, \\ -\infty & \text{otherwise.} \end{cases}$$

3.2 Unified Competitive Learning SMoE (USMoE)

Unified Competitive Score. As illustrated in Figure 2, the Token Choice approach selects and promotes the best expert by comparing the similarity scores between a token and the available experts. Given a representation $\mathbf{x} \in \mathbb{R}^{b \times l \times d}$, which is the output of the multi-head attention (MHA) layer, and a router matrix $W_e \in \mathbb{R}^{d \times n}$, where n represents the number of experts, the similarity score s_t in the Token Choice approach is computed as:

$$s_t = \text{softmax}(W_e \cdot x, d = -1) \qquad (3)$$

where d denotes the dimension along which the softmax function is applied. In contrast, the Expert Choice approach selects and promotes the best tokens by comparing the similarity scores between an expert and the available tokens. The similarity score s_e in the Expert Choice approach is computed as:

$$s_e = \text{softmax}(W_e \cdot x, d = 1) \qquad (4)$$

While Expert Choice ensures an equal distribution of tokens across experts, Token Choice guarantees that each token is fairly processed by the model. To leverage the strengths of both approaches, we introduce the Unified Competitive Score, a soft combination of the Token Choice and Expert Choice scores, defined as:

$$s_u = \alpha \cdot s_e + (1 - \alpha) \cdot s_t \qquad (5)$$

where $\alpha \in [0, 1]$ is a tunable control parameter that can be adjusted based on the data. In practice, we find that $\alpha \approx 0.5$ is an appropriate choice.

Unified Competitive Mechanism. Token Choice selects experts using horizontal competitive learning, where each token is assigned to the most similar expert. This approach performs well when there is a highly relevant expert for a given token. However, its effectiveness diminishes when the similarity scores between the token and all experts are low. In contrast, Expert Choice selects tokens using vertical competitive learning, where each expert is assigned the most similar tokens. This approach is effective when the similarity score distribution across experts is well-structured. However, this condition is difficult to achieve due to the issue of representation collapse (Chi et al., 2022b). As a result, Expert Choice may either discard important tokens or lead to multiple experts selecting the same token. USMoE addresses these challenges by introducing the *Unified Competitive Mechanism*, which treats expert and token selection as a joint (expert, token) pairing process. As demonstrated in Algorithm 1, we first flatten the similarity matrix and then select the top-N highest-scoring pairs, representing the most similar expert-token combinations.

Let the ordered array $(S_1^t, S_2^t, \ldots, S_N^t)$ represent the top-N token scores selected using the Token Choice approach, where $S_1^t \leq S_2^t \leq \cdots \leq S_N^t$. Similarly, let $(S_1^e, S_2^e, \ldots, S_N^e)$ denote the top-N token scores selected using the Expert Choice approach, where $S_1^e \leq S_2^e \leq \cdots \leq S_N^e$. Finally, let $(S_1^u, S_2^u, \ldots, S_N^u)$ represent the top-N token scores selected using the Unified Competitive Score approach, where $S_1^u \leq S_2^u \leq \cdots \leq S_n^u$. We derive the following inequalities, with a detailed proof provided in the Appendix A.1:

$$\begin{aligned} S_i^t &\leq S_i^u, \quad \forall i \in [1, N] \\ S_i^e &\leq S_i^u, \quad \forall i \in [1, N] \end{aligned} \tag{6}$$

Based on Inequality 6, the *Unified Competitive Mechanism* ensures that the selected expert-token pairs are at least as optimal as those chosen by the Token Choice approach. Moreover, this mechanism addresses the limitations of Expert Choice, which may drop important tokens, and Token Choice, which may select irrelevant tokens.

3.3 Theoretical Analysis for Representation Collapse of Sparse Mixture of Experts

Following (Chi et al., 2022a) and (Do et al., 2023b), we illustrate the representation collapse issue using the Jacobian matrix approach. Specifically, the Jacobian matrix of the SMoE with respect to $x \in \mathbb{R}^{b \times d}$ is given as:

$$\boldsymbol{J_{SMoE}} = \mathcal{S}(x)_k \boldsymbol{J}^{\text{FFN}} + \textstyle\sum_{j=1}^{n} \mathcal{S}(x)_k \left(\delta_{kj} - S_j\right) \boldsymbol{E}(x)_i \boldsymbol{e}_j^\top$$

$$\implies \boldsymbol{J_{SMoE}} = \mathcal{S}(x)_k \boldsymbol{J}^{\text{FFN}} + \sum_{j=1}^{n} \boldsymbol{c}_j \boldsymbol{e}_j^\top, \tag{7}$$

where $\boldsymbol{c}_j = \mathcal{S}(x)_k \left(\delta_{kj} - S_j\right) \boldsymbol{E}(x)_i$. The first part of Equation 7, $\mathcal{S}(x)_k \boldsymbol{J}^{\text{FFN}}$, represents the contribution from the input token and experts to the final output. The second part, (2) $\sum_{j=1}^{n} \boldsymbol{c}_j \boldsymbol{e}_j^\top$ relates to learning an improved gating function to minimize task loss. Furthermore, Equation 7 should be updated as a linear combination of expert embeddings. Due to $n << d$ in practice, the above equation illustrates the representation.

Algorithm 1 USMoE Layer

Require: $X \in \mathbb{R}^{B \times L \times D}$, router weights $R \in \mathbb{R}^{D \times N}$, experts, controlling factor α
Ensure: Output Y

Compute logits:
  $\text{logits} \leftarrow X \cdot R$ ▷ Dot product Similarity
Compute token choice score:
  $\text{tc\_score} \leftarrow \text{softmax}(\text{logits}, \text{axis} = -1)$
Compute expert choice score:
  $\text{ex\_score} \leftarrow \text{softmax}(\text{logits}, \text{axis} = 1)$
Compute unified score:
  $U \leftarrow \alpha \cdot \text{ex\_score} + (1 - \alpha) \cdot \text{tc\_score}$
Reshape score:
  $U \leftarrow \text{reshape}(U, B, -1)$
Compute top-N indices and values:
  $\text{topn\_val}, \text{topn\_idx} \leftarrow$
  $\text{TopK}(U, \text{topn}, \text{dim} = 1)$
Compute output using SMoE:
  $Y \leftarrow$
  $\text{SMoE}(X, \text{experts}, \text{topn\_val}, \text{topn\_idx})$
return Y

Given $\mathcal{S}(x) = \alpha \times \mathcal{S}_e(x) + (1-\alpha) \times \mathcal{S}_t(x)$, where $\mathcal{S}_e(x)$ and $\mathcal{S}_t(x)$ represent the similarity score functions for expert choice and token choice, respectively, the Jacobian matrix of USMoE with respect to $x \in \mathbb{R}^{b \times d}$ is expressed as:

$$\begin{aligned} \boldsymbol{J_U} = \mathcal{S}(x)_k \boldsymbol{J}^{\text{FFN}} &+ \sum_{j=1}^{n} \mathcal{S}_e(x)_k \left(\delta_{kj} - \mathcal{S}_e(x)_j\right) \boldsymbol{E}(x)_k \boldsymbol{e}_j^\top \\ &+ \sum_{j=1}^{n} \mathcal{S}_t(x)_k \left(\delta_{kj} - \mathcal{S}_t(x)_j\right) \boldsymbol{E}(x)_k \boldsymbol{e}_j^\top. \end{aligned}$$

$$\implies \boldsymbol{J_U} = J_1 + \sum_{j=1}^{n} c_j e_j^\top + \sum_{j=1}^{n} d_j e_j^\top \tag{8}$$

where

$$J_1 = \mathcal{S}(x)_k \boldsymbol{J}^{\text{FFN}}, \tag{9}$$

$$c_j = \mathcal{S}_e(x)_k \left(\delta_{kj} - \mathcal{S}_e(x)_j\right) \boldsymbol{E}(x)_k \boldsymbol{e}_j^\top, \tag{10}$$

$$d_j = \mathcal{S}_t(x)_k \left(\delta_{kj} - \mathcal{S}_t(x)_j\right) \boldsymbol{E}(x)_k. \tag{11}$$

Similar to the Jacobian matrix of SMoE as Equation 7, the Jacobian matrix of USMoE also consists of two terms: (1) J_1, which depends on the input token and experts for the final output; and (2) $\sum_{j=1}^{2n} o_j \boldsymbol{e}_j^\top$ indicates to learn better gating function to minimize the task loss. Since $2n >> n$,

Token Choice	Expert 1	Expert 2	Expert 3	Expert 4
Token1	0.2	**0.3**	0.2	0.3
Token2	**0.5**	0.4	0.1	0.0
Token3	**0.7**	0.2	0.0	0.1
Token4	0.3	0.0	**0.6**	0.1

Expert Choice	Expert 1	Expert 2	Expert 3	Expert 4
Token1	0.2	0.3	0.2	**0.3**
Token2	0.5	**0.4**	0.1	0.0
Token3	**0.7**	0.2	0.0	0.1
Token4	0.3	0.0	**0.6**	0.1

Flatten
0.2
0.3
0.2
0.3
0.5
0.4
0.1
0.0
0.7
0.2
0.0
0.1
0.3
0.0
0.6
0.1

USMoE	Expert 1	Expert 2	Expert 3	Expert 4
Token1	0.2	0.3	0.2	0.3
Token2	**0.5**	**0.4**	0.1	0.0
Token3	**0.7**	0.2	0.0	0.1
Token4	0.3	0.0	**0.6**	0.1

Figure 2: An illustration of our USMoE framework, which enhances existing SMoE models through a Unified Competitive Learning approach. The method first flattens the score matching between tokens and experts into a **1D representation**, then selects the **Top-N** best pairs, and finally maps the results back to the original 2D matching scores. Best viewed in color.

USMoE is more effective than SMoE in addressing the representation collapse issue.

4 Experiments

We evaluate our method in two scenarios: (1) without additional training and (2) with training. For the first scenario, inspired by (Li and Zhou, 2024), we test our method as a plug-in framework on well-trained SMoE models, including **OLMoE-1B-7B** (Muennighoff et al., 2024):, **DeepSeekMoE-16B** (Dai et al., 2024), **Qwen1.5-MoE-A2.7B** (Team, 2024). We evaluate performance on a subset of tasks from the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023), which covers key downstream applications for sentence embeddings, including Classification, Clustering, Pair Classification, Re-ranking, Retrieval, Semantic Textual Similarity (STS), and Summarization. Following the MTEB evaluation framework, we use Accuracy for Classification, V-Measure for Clustering, Average Precision for Pair Classification, Mean Average Precision for Re-ranking, nDCG for Retrieval, and Spearman's correlation for STS and Summarization.

For the second scenario, we perform language model pre-training on diverse datasets, including Enwik8, Text8 (Mahoney, 2011), Wikitext-103 (Merity et al., 2017), and One Billion Words (Chelba et al., 2014). To assess performance, we fine-tune the pre-trained models on various downstream benchmarks.

4.1 Experiment Setting

We evaluate our method against two conventional SMoE approaches: (1) the Token Choice approach and (2) the Expert Choice approach, under three different settings: (1) without additional training, (2) pre-training, and (3) fine-tuning.

Without Training. We evaluate our approach on three state-of-the-art Sparse Mixture of Experts (MoE) models: (1) *OLMoE-1B-7B* (Muennighoff et al., 2024), which has 7B parameters, 16 layers, and 64 experts per layer; (2) Qwen1.5-MoE-A2.7B (Team, 2024), comprising 7B parameters, 24 layers, and 60 experts per layer; and *DeepSeekMoE-16B* which consists of 16B parameters, 28 layers, and 64 experts per layer. We evaluate these models on the Massive Text Embedding Benchmark (MTEB) without additional fine-tuning. Additionally, we compare methods both with and without prompts, including PromptEOL (Jiang et al., 2024c). Inspired by (Li and Zhou, 2024), we also compare our method with an approach that uses the similarity score between router and expert embeddings as the hidden representation, which we refer to as "Router Embedding" or simply "Router" Additionally, we evaluate against MoEE (Li and Zhou, 2024), which leverages both Router Embedding and the hidden representation of the SMoE model as embeddings.

Pre-training. To assess the effectiveness of our method, we compare USMoE with the Token Choice approaches, including SMoE (Jiang et al., 2024b), SMoE-Dropout (abbreviated as "SMoE-DR"), XMoE (Chi et al., 2022b), and Stable-MoE (Dai et al., 2022a), as well as the Expert

Choice approach (Zhou et al., 2022a) for pre-training and fine-tuning tasks. We follow the approach of Chen et al. (2023b) and use a base Transformer-XL (Dai et al., 2019a) with four decoder layers. We train both base and large-scale versions of Transformer-XL on four datasets (Enwik8, Text8, Wikitext-103, and One Billion Words) for 100k iterations, following the implementation in (Chen et al., 2023b).

Fine-tuning. We fine-tune the pre-trained weights for text classification tasks, including `SST-2` (Socher et al., 2013), `SST-5` (Socher et al., 2013), `IMDB` (Maas et al., 2011), and `BANKING77` (Casanueva et al., 2020). More implementation details and additional results are provided in the Appendix A.

Model	Task	Router	TC	EC	MoEE	USMoE
OLMoE-1B-7B	Classification	43.1	57.7	56.2	51.7	**59.2**
	Clustering	16.2	24.8	26.9	23.2	**30.5**
	PairClassification	53.5	62.0	58.9	66.0	**66.8**
	Reranking	41.7	51.3	51.0	53.2	**54.7**
	STS	49.4	63.5	44.2	67.8	**71.1**
	Summarization	25.6	28.9	29.7	30.4	**30.9**
	Average	38.3	48.0	44.5	48.7	**52.2**
Qwen1.5-MoE-A2.7B	Classification	48.8	58.0	35.2	54.0	**59.2**
	Clustering	14.3	34.2	29.2	30.1	**35.8**
	PairClassification	51.9	60.5	56.0	60.3	**63.4**
	Reranking	41.0	46.6	45.0	51.1	**53.7**
	STS	48.3	50.1	50.0	64.3	**66.3**
	Summarization	27.0	23.0	21.9	27.3	**48.4**
	Average	38.6	45.4	39.6	47.9	**54.5**
DeepSeekMoE-16B	Classification	48.6	56.4	55.4	53.0	**57.3**
	Clustering	17.8	29.0	20.3	28.5	**31.9**
	PairClassification	57.4	59.8	53.8	63.3	**65.3**
	Reranking	43.8	45.7	40.9	50.6	**52.1**
	STS	52.8	49.0	37.1	63.4	**66.0**
	Summarization	29.1	24.4	25.7	29.2	**30.9**
	Average	41.6	44.0	38.9	48.0	**50.6**

Table 1: Performance comparison of USMoE, Token Choice (TC), Expert Choice (EC), and MoEE across across MTEB Tasks **with PromptEOL (Jiang et al., 2023)**. The best result for each row is highlighted in **bold**.

4.2 Without Training

Our method consistently demonstrates performance improvements across a range of MTEB tasks in two scenarios: (1) with PromptEOL (Jiang et al., 2023) (denoted as "with prompts" for brevity), as shown in Table 1, and (2) without prompts, as shown in Table 2. Detailed results for datasets under each task type are provided in Appendix A. USMoE outperforms both the Expert Choice and Token Choice approaches in most cases, underscoring the complementary nature of these two methods.

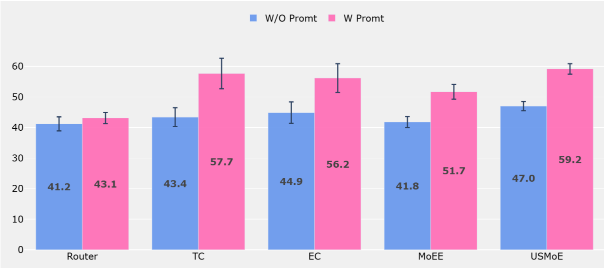
Figure 1 illustrates the effectiveness of our method across various MTEB tasks using the *OLMoE-1B-7B* model, in comparison to both the Expert Choice and Token Choice approaches. For tasks evaluated with prompts, USMoE achieves the highest average performance across models, showing notable improvements of **4.2%**, **9.1%**, and **6.6%** for *OLMoE-1B-7B*, *Qwen1.5-MoE-A2.7B*, and *DeepSeekMoE-16B*, respectively, without requiring any additional training, as detailed in Table 1. Notably, *DeepSeekMoE-16B* demonstrates a significant improvement from 50.1% (Token Choice) to 64.3% (USMoE), reflecting a 14.2% gain in the STS task.

For tasks evaluated without prompts, USMoE proves even more effective at enhancing the Token Choice approach, delivering notable gains of **9.3%**, **7.0%**, and **8.0%** for *OLMoE-1B-7B*, *Qwen1.5-MoE-A2.7B*, and *DeepSeekMoE-16B*, respectively, across MTEB tasks without additional training. Specifically, *OLMoE-1B-7B* achieves a remarkable improvement from 24.1% (Token Choice) to 47.5% (USMoE), representing a 23.4% gain. This trend persists across *Qwen1.5-MoE* and *OLMoE*, where USMoE consistently outperforms both the Token Choice and Expert Choice approaches.

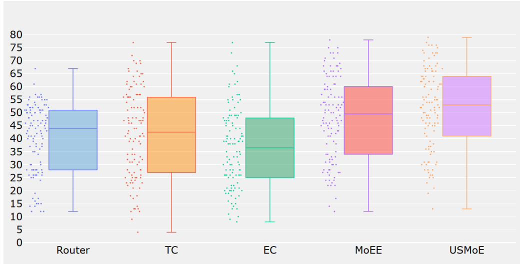
Interestingly, Router Embedding is less affected by prompting on the Classification dataset, as shown in Figure 3a, while Token Choice, Expert Choice, and USMoE (ours) achieve significant performance improvements in the prompting setting. Figure 3a also demonstrates that our method is not only more effective but also more stable than the baselines, as indicated by a lower *standard deviation*. Additionally, Figure 3b illustrates the distribution of our method across MTEB tasks in both prompted and non-prompted scenarios. Overall, our approach outperforms the baselines in terms of performance while exhibiting lower variance across multiple tasks and different runs.

4.3 Pre-training Result

Base models training. Table 3 summarizes the pre-training results across four datasets (`enwik8`, `text8`, `WikiText-103`, and `One Billion Words`). USMoE consistently outperforms the expert choice baseline and token choice routing methods such as XMoE (Chi et al., 2022a) and StableMoE (Dai et al., 2022b) on all four datasets. The strength of USMoE lies in its inference efficiency, achieved by leveraging fewer experts. Notably, on `text8`, USMoE surpasses SMoE while utilizing only one and a half experts. Additionally, it outperforms SMoE-Dropout (which employs two experts) on `One Billion Words`, lowering perplexity from



(a) Performance comparison of USMoE, Token Choice (TC), and Expert Choice (EC) on Classification Task.



(b) Results distribution of USMoE, Token Choice (TC), Expert Choice (EC), and MoEE across MTEB Tasks

Figure 3: Illustration of comparing the performance of USMoE, Token Choice (TC), Expert Choice (EC), and MoEE across MTEB tasks and three SMoE models. Each benchmark is run 10 times, reporting both the mean and standard deviation to highlight the performance and stability of our method compared to the baselines.

Model	Task	Router	TC	EC	MoEE	USMoE
OLMoE-1B-7B	Classification	41.2	43.4	44.9	41.8	**47.0**
	Clustering	13.7	14.7	12.0	14.5	**16.9**
	PairClassification	45.3	39.1	35.5	45.7	**50.1**
	Reranking	37.5	37.4	35.3	39.5	**43.1**
	STS	39.9	24.1	18.2	39.9	**47.5**
	Summarization	28.4	20.9	21.1	29.8	**30.8**
	Average	34.3	29.9	27.8	35.2	**39.2**
Qwen1.5-MoE-A2.7B	Classification	43.8	50.3	25.5	47.7	**51.2**
	Clustering	13.6	27.4	23.2	25.2	**26.7**
	PairClassification	45.9	46.9	43.4	51.5	**53.1**
	Reranking	39.6	45.3	41.6	48.5	**48.2**
	STS	38.8	38.0	35.6	51.8	**54.8**
	Summarization	28.3	13.4	15.1	**31.2**	29.7
	Average	35.0	36.9	30.7	42.6	**43.9**
DeepSeekMoE-16B	Classification	43.4	46.6	44.7	44.4	**46.8**
	Clustering	13.4	18.1	13.5	17.8	**21.9**
	PairClassification	45.5	40.9	37.1	46.1	**51.9**
	Reranking	38.5	38.9	35.1	42.2	**45.7**
	STS	37.7	26.3	23.3	40.2	**45.9**
	Summarization	24.9	22.0	18.5	24.4	**28.4**
	Average	33.9	32.1	28.7	35.9	**40.1**

Table 2: Performance comparison of USMoE, Token Choice (TC), Expert Choice (EC), and MoEE across MTEB Tasks **without prompts** and models. The best result for each row is highlighted in **bold**.

Transformer-XL(20M)		Enwik8	Text8	WikiText-103	lm1b
USMoE (Top$k$=2)		**1.18**	**1.20**	**29.20**	**56.90**
(Top$k$=1.5)		1.19	1.28	30.67	57.55
TC (Top$k$=2)	SMoE	1.20	1.29	30.16	58.00
	SMoE-DR	1.56	1.56	58.37	93.17
	XMoE	1.21	1.28	30.34	58.33
	StableMoE	1.20	1.28	29.97	58.25
EC (Top$k$=2)		1.18	1.24	29.83	58.60

Table 3: Performance comparison of USMoE, Token Choice (TC), and Expert Choice (EC) across multiple datasets, with BPC on the Enwik8 and Text8 test sets, and perplexity on the WikiText-103 and One Billion Word test sets. Lower values are better, with the best results highlighted in **bold**.

93.17 to **57.55** with the same reduced expert count. Furthermore, with only one and a half experts, USMoE **reduces FLOPs by 14%** compared to SMoE and SMoE-Dropout, which rely on two experts, all while maintaining strong performance.

Large models training. USMoE not only delivers strong performance in base model training but also remains highly competitive at a large scale. Table 4 presents perplexity (PPL) results on the `WikiText-103` and `One Billion Words` datasets using a large Transformer-XL model with *15 SMoE layers*, *16 experts*, and *420M parameters*. The performance gap between USMoE and the baselines becomes even more pronounced at this scale, highlighting its strong scalability with increasing model complexity. Regardless of backbone size or the number of activated experts, USMoE consistently outperforms all baselines, demonstrating its effectiveness in scaling up large language models.

Transformer-XL(420M)	WikiText-103			lm1b		
Top$k$	TC	EC	USMoE	TC	EC	USMoE
1	31.70	35.52	**25.48**	58.65	65.43	**56.90**
2	22.42	23.30	**22.06**	44.56	43.39	**40.53**
4	23.57	23.60	**22.65**	45.52	43.70	**40.90**
8	24.20	24.37	**22.88**	46.36	44.22	**43.24**

Table 4: Large Scale performance comparison of USMoE, Token Choice (TC), and Expert Choice (EC) across multiple datasets, with perplexity on the WikiText-103 and One Billion Word test sets. Lower values are better, with the best results highlighted in **bold**.

4.4 Fine-tuning Result

Fine-tuning. We report the results of the fine-tuning experiment on the `SST-2`, `SST-5`, `IMDB`, and `BANKING77` datasets in Table 5, using Transformer-XL pre-trained on `enwik8`. Overall, USMoE consistently achieves higher accuracy compared to other baselines across all datasets. The results demonstrate that our method is not only effective for pre-training tasks but also performs effectively on existing pre-trained models.

Transformer-XL(20M)		FLOPs(x10^{10})	SST-2	SST-5	IMDB	BANKING77
USMoE (Top$k$=2)		7.7620	81.5	**40.1**	**88.5**	**87.8**
(Top$k$=1.5)		**6.6753**	**83.8**	39.6	88.3	83.0
TC (Top$k$=2)	SMoE	7.7620	77.1	35.1	84.4	69.2
	SMoE-DR		78.6	34.4	83.5	66.7
	XMoE		76.7	35.3	83.3	67.4
	StableMoE		77.7	34.3	83.9	60.8
EC (Top$k$=2)		7.7620	81.5	39.3	88.0	75.6

Table 5: Accuracy performance comparison of USMoE, Token Choice (TC), and Expert Choice (EC) after fine-tuned on various datasets. Higher is better, best results are in **bold**.

Model	Dataset	α					
		0.0	0.3	0.5	0.7	0.9	1.0
DeepSeekMoE-16B	Emotion	27.4	27.1	**27.8**	27.6	27.7	26.5
	Toxic	**60.4**	60.0	60.1	56.8	57.3	58.1
	Tweet	51.9	**53.2**	52.5	53.3	52.9	49.5

Table 7: Performance comparison of DeepSeekMoE-16B across different classification datasets with varying α values. Higher is better; best results are in **bold**.

4.5 Ablation Studies

We investigate the effectiveness and robustness of USMoE to the different hyper-parameter settings.

4.5.1 Competitive Learning Strategy Comparison

Model	Dataset	TC	EC	USMoE-Sequence	USMoE-Batch
DeepSeekMoE-16B	Emotion	27.4	26.5	**27.8**	27.4
	Toxic	**60.4**	58.1	60.1	59.2
	Tweet	51.9	49.5	**52.5**	51.7

Table 6: Competitive Learning Strategy comparison of USMoE, Token Choice (TC), and Expert Choice (EC) on the classification task. Higher values are better, with the best results highlighted in **bold**.

The Unified Competitive Mechanism is implemented using two approaches: (1) a sequence-based method that compares all tokens within a sequence (referred to as "USMoE-Sequence") and (2) a batch-based method that compares all tokens within a batch or mini-batch (referred to as "USMoE-Batch"). We evaluate both approaches on the Classification task, with results presented in Table 6. The findings indicate that both methods outperform the Expert Choice and Token Choice approaches, demonstrating the effectiveness of our method. Notably, the sequence-based approach achieves superior performance in the Classification task, as it ensures that no important tokens are missed within a sequence - an assurance that the batch/mini-batch implementation may not always provide.

4.5.2 Robustness to the controlling factor α

The Unified Competitive Score (α) enables the model to adjust its scoring mechanism, either favoring a diverse set of experts per sequence or distributing experts more evenly across tokens. We evaluate the robustness of the controlling factor α on the classification task using the *DeepSeekMoE-16B* model, with results presented in Table 7. When $\alpha = 0.0$, the scoring mechanism aligns with Token Choice, while at $\alpha = 1.0$, it follows Expert Choice. Overall, USMoE demonstrates strong effectiveness within the range of $\alpha \in (0.3, 0.7)$, striking a balance between expert diversity and token importance. This range provides an optimal trade-off between enforcing SMoE's competition policy and enhancing traditional approaches for the task, leading to superior overall performance. Notably, all tested α configurations outperform the Expert Choice approach ($\alpha = 1.0$).

Summary and hyper-parameter configuration guidelines. Our experiments demonstrate that USMoE not only excels in pre-training and transfer learning but also achieves superior generalization compared to traditional approaches, all without requiring additional training. Ablation studies in Section 4.5 indicate that USMoE is highly effective within the range of $\alpha \in (0.3, 0.7)$. This finding provides a practical guideline for efficiently tuning USMoE through cross-validation.

5 Conclusion

In this research, we reformulate Token Choice and Expert Choice Sparse Mixture of Experts from a competitive learning perspective, highlighting their limitations. Building on this analysis, we introduce Unified Competitive Learning SMoE (USMoE) - a novel and efficient framework that enhances SMoE through a unified competitive learning approach.
We theoretically prove that USMoE achieves superior expert selection compared to traditional methods, effectively improving expert learning capacity while mitigating expert collapse. As a result, USMoE learns more robust expert representations and overcomes the representation collapse issues commonly observed in conventional SMoE training.
Experiments across both training-free and training-based settings (including pre-training and fine-tuning) demonstrate that USMoE enables more efficient and effective training and inference compared to state-of-the-art routing strategies.

Limitations

Our study focuses on enhancing the efficiency and effectiveness of Large Language Models (LLMs) through SMoE. Our approach proves effective in both training and non-training settings. While the results are promising, our pre-training experiments were constrained by computational resources, limiting us to medium-scale datasets and a base Transformer-XL model. Consequently, further empirical evaluation is required to assess the scalability of USMoE beyond 100B parameters and compare it with other SMoE strategies in modern LLMs.

Ethics Statement

Despite encouraging results, training large-scale LLMs remains highly resource-intensive, requiring careful management of computational costs. Additionally, our study relies on web-sourced data, which is known to contain gender and racial biases, highlighting the need for further efforts to mitigate these issues. Lastly, while our work represents a significant step toward advancing LLMs development, it also emphasizes the importance of robust regularization to prevent potential misuse in harmful applications.

References

Junhyeok Ahn and Luis Sentis. 2021. Nested mixture of experts: Cooperative and competitive learning of hybrid dynamical system. *Preprint*, arXiv:2011.10605.

Richard A. Andersen, John C. Eccles, and Thomas A. Sears. 1969. Participation of inhibitory burst neurons in the generation of oculomotor commands. *Journal of Neurophysiology*, 32(3):356–372.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giri Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Ves Stoyanov. 2022. Efficient large scale language modeling with mixtures of experts. *Preprint*, arXiv:2112.10684.

Shenghong Cai, Yiqun Zhang, Xiaopeng Luo, Yiu-Ming Cheung, Hong Jia, and Peng Liu. 2024. Robust categorical data clustering guided by multi-granular competitive learning. In *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, pages 288–299.

Yinqiong Cai, Yixing Fan, Keping Bi, Jiafeng Guo, Wei Chen, Ruqing Zhang, and Xueqi Cheng. 2023. Came: Competitively learning a mixture-of-experts model for first-stage retrieval. *Preprint*, arXiv:2311.02834.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. *Preprint*, arXiv:1312.3005.

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

Tianlong Chen, Zhenyu Zhang, AJAY KUMAR JAISWAL, Shiwei Liu, and Zhangyang Wang. 2023b. Sparse MoE as the New Dropout: Scaling Dense and Self-Slimmable Transformers. In *The Eleventh International Conference on Learning Representations*.

Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. 2022. Towards Understanding the Mixture-of-Experts Layer in Deep Learning. In *Advances in Neural Information Processing Systems*.

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. 2022a. On the Representation Collapse of Sparse Mixture of Experts. In *Advances in Neural Information Processing Systems*.

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. 2022b. On the representation collapse of sparse mixture of experts. *Preprint*, arXiv:2204.09179.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *Preprint*, arXiv:1904.10509.

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. 2022a. Stablemoe: Stable routing strategy for mixture of experts. *Preprint*, arXiv:2204.08396.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. 2022b. StableMoE: Stable Routing Strategy for Mixture of Experts. In

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7085–7095, Dublin, Ireland. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019a. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019b. Transformer-xl: Attentive language models beyond a fixed-length context. *Preprint*, arXiv:1901.02860.

Giang Do, Hung Le, and Truyen Tran. 2024a. 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. 2023a. Hyperrouter: Towards efficient training and inference of sparse mixture of experts. *Preprint*, arXiv:2312.07035.

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

Giang Do, Kha Pham, Hung Le, and Truyen Tran. 2024b. On the effectiveness of discrete representations in sparse mixture of experts. *Preprint*, arXiv:2411.19402.

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, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 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.

Le Hou, Richard Yuanzhe Pang, Tianyi Zhou, Yuexin Wu, Xinying Song, Xiaodan Song, and Denny Zhou. 2022. Token dropping for efficient bert pretraining. *Preprint*, arXiv:2203.13240.

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, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024a. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024b. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2023. Scaling sentence embeddings with large language models. *Preprint*, arXiv:2307.16645.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024c. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196, Miami, Florida, USA. Association for Computational Linguistics.

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

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *Preprint*, arXiv:2307.10169.

Samuel Kaski and Teuvo Kohonen. 1994. Winner-take-all networks for physiological models of competitive learning. *Neural Networks*, 7(6):973–984. Models of Neurodynamics and Behavior.

Jakub Krajewski, Jan Ludziejewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, Marek Cygan, and Sebastian Jaszczur. 2024. Scaling laws for fine-grained mixture of experts. *Preprint*, arXiv:2402.07871.

Ziyue Li and Tianyi Zhou. 2024. Your mixture-of-experts llm is secretly an embedding model for free. *Preprint*, arXiv:2410.10814.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2025. Rho-1: Not all tokens are what you need. *Preprint*, arXiv:2404.07965.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Matt Mahoney. 2011. Large text compression benchmark.

James L. McClelland and David E. Rumelhart. 1987. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 2: Psychological and Biological Models*. MIT Press, Cambridge, MA.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer Sentinel Mixture Models. In *International Conference on Learning Representations*.

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, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. 2024. Olmoe: Open mixture-of-experts language models. *Preprint*, arXiv:2409.02060.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. *Preprint*, arXiv:2210.07316.

Quang Pham, Giang Do, Huy Nguyen, TrungTin Nguyen, Chenghao Liu, Mina Sartipi, Binh T. Nguyen, Savitha Ramasamy, Xiaoli Li, Steven Hoi, and Nhat Ho. 2024a. Competesmoe – effective training of sparse mixture of experts via competition. *Preprint*, arXiv:2402.02526.

Quang Pham, Giang Do, Huy Nguyen, TrungTin Nguyen, Chenghao Liu, Mina Sartipi, Binh T. Nguyen, Savitha Ramasamy, Xiaoli Li, Steven Hoi, and Nhat Ho. 2024b. Competesmoe – effective training of sparse mixture of experts via competition. *Preprint*, arXiv:2402.02526.

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.

David E. Rumelhart and David Zipser. 1985. Feature discovery by competitive learning. *Cognitive Science*, 9(1):75–112.

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

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017b. 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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Rupesh K Srivastava, Jonathan Masci, Sohrob Kazerounian, Faustino Gomez, and Jürgen Schmidhuber. 2013. Compete to compute. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Costas Stefanis and Herbert H. Jasper. 1969. Interneuronal inhibition in the cerebral cortex. i. a study of inhibition between pyramidal cells. *Journal of Neurophysiology*, 32(3):389–403.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hongqiu Wu, Hai Zhao, and Min Zhang. 2021. Not all attention is all you need. *Preprint*, arXiv:2104.04692.

Qinlin Zhao, Jindong Wang, Yixuan Zhang, Yiqiao Jin, Kaijie Zhu, Hao Chen, and Xing Xie. 2024. Competeai: Understanding the competition dynamics in large language model-based agents. *Preprint*, arXiv:2310.17512.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. 2022a. Mixture-of-experts with expert choice routing. *Preprint*, arXiv:2202.09368.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, zhifeng Chen, Quoc V Le, and James Laudon. 2022b. Mixture-of-Experts with Expert Choice Routing. In *Advances in*

Neural Information Processing Systems, volume 35, pages 7103–7114. Curran Associates, Inc.

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

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. *Preprint*, arXiv:2202.08906.

A Appendix

Supplementary Material for "Sparse Mixture of Experts as Unified Competitive Learning"

This document is organized as follows: Appendix A.1 provides the theoretical proof supporting the discussion in Section 3.2. Following this, Appendix A.3 offers a detailed analysis of the SMoE router and explains why our method outperforms the baseline approaches. Appendix A.2 presents supplementary experimental results, and Appendix A.4 describes the implementation details in full.

A.1 Theoretical Proof for Section 3.2

Let $x \in \mathbb{R}^{N\times d}$ represent an input token embedding and $\mathcal{E} \in \mathbb{R}^{d\times n}$ represent the expert embeddings, where d is the dimension of the SMoE model and n is the number of experts. The similarity score S between the input token and the expert embeddings is computed using a dot product:

$$S = x \cdot \mathcal{E}. \tag{12}$$

Since the softmax function is monotonic, the selection of top-k values remains unchanged under softmax transformation. Formally, we have:

$$\mathrm{TopK}(\mathrm{softmax}(S), k) = \mathrm{TopK}(S, k). \tag{13}$$

By setting $k = 1$, the top similarity score for a given token using the **Token Choice** approach is:

$$S_j^t = \max(S_{ik}), \quad \forall k \in [1, n]. \tag{14}$$

Since this operation is performed independently for each input token i, we extend this to all tokens q in the batch:

$$S_j^t \leq \max(S_{qk}), \quad \forall k \in [1, n], \quad \forall q \in [1, N]. \tag{15}$$

By definition, the **Unified Competitive Mechanism** approach selects the highest similarity score across all tokens and experts:

$$S_j^u = \max(S_{qk}), \quad \forall k \in [1, n], \quad \forall q \in [1, N]. \tag{16}$$

Thus, we establish the first inequality:

$$S_i^t \leq S_i^u, \quad \forall i \in [1, N]. \tag{17}$$

Similarly, in the **Expert Choice** approach, each expert selects the best matching token, leading to:

$$S_i^e \leq S_i^u, \quad \forall i \in [1, N]. \tag{18}$$

Therefore, we conclude:

$$S_i^t \leq S_i^u, \quad S_i^e \leq S_i^u, \quad \forall i \in [1, N]. \tag{19}$$

This completes the proof. □

A.2 Additional Results

We provide a detailed evaluation of three state-of-the-art SMoE models: **OLMoE-1B-7B** (Table 8), **Qwen1.5-MoE-A2.7B** (Table 9), and **DeepSeekMoE-16B** (Table 10). Our results demonstrate the effectiveness of our method across various models and prompts, comparing its performance against baseline approaches such as Token Choice (TC) and Expert Choice (EC).

A.3 In-depth Analysis

We visualize the router behavior of USMoE in Figure 6 and contrast it with the router behaviors of the Token Choice approach (Figure 4) and the Expert Choice approach (Figure 5). Notably, the router in the **OLMoE-1B-7B** model exhibits a strong preference for specific experts. For instance, in the *Emotion Classification* task, Experts 8, 30, and 58 are consistently prioritized in both the Token Choice and Expert Choice approaches. This bias limits the model's adaptability and effectiveness for downstream tasks. USMoE tackles this challenge by introducing the **Unified Competitive Mechanism**, which promotes more balanced and diverse expert selections, as illustrated in Figure 6. This enhancement enables USMoE to outperform the baselines on the *Emotion Classification* task.

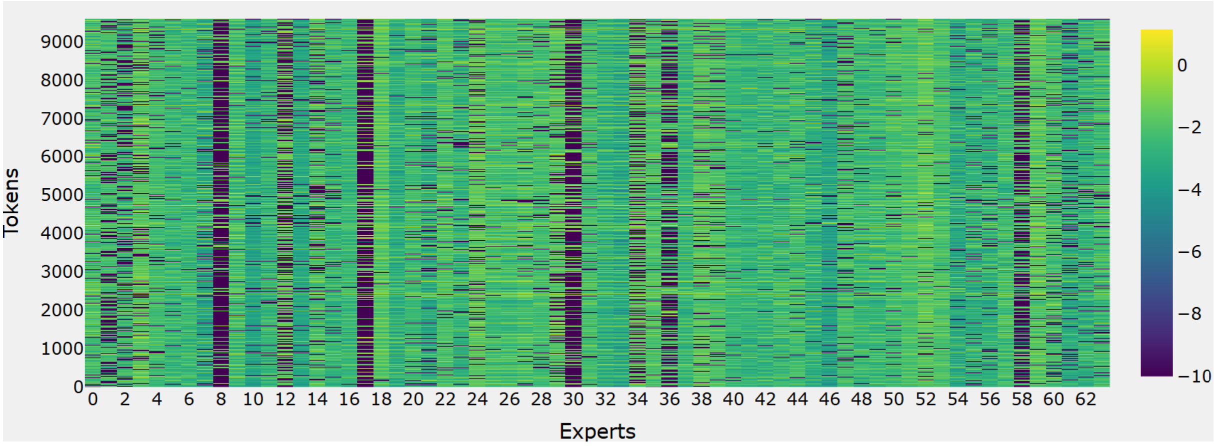


Figure 4: **Token Choice Router** visualization for the **OLMoE-1B-7B** model on the *Emotion Classification* task. The scores of selected experts are replaced with *-10.0* (lower than the minimum score) to enhance visualization. Best viewed in color.

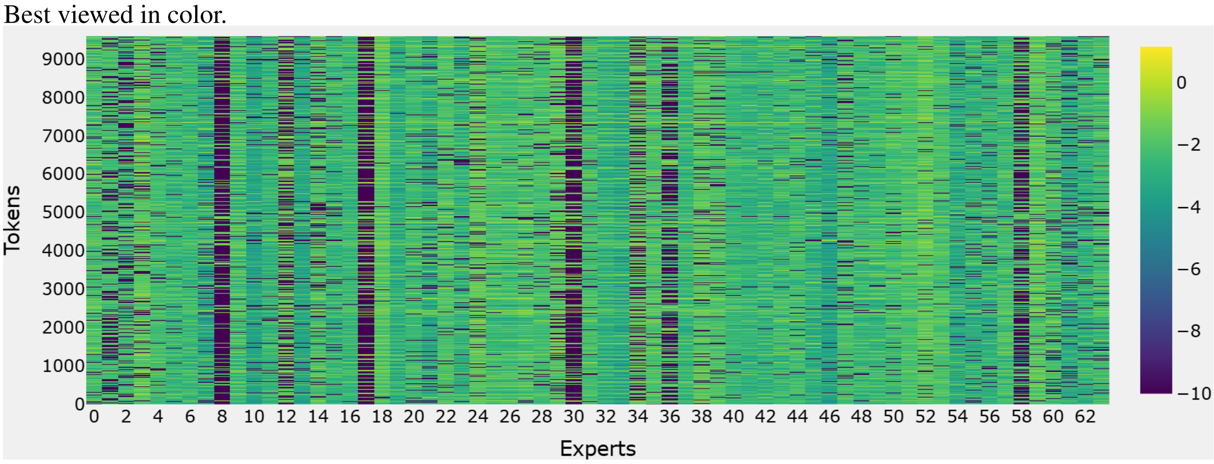


Figure 5: **Expert Choice Router** visualization for the **OLMoE-1B-7B** model on the *Emotion Classification* task. The scores of selected experts are replaced with *-10.0* (lower than the minimum score) to enhance visualization. Best viewed in color.

We track the number of unique experts utilized by the **OLMoE-1B-7B** model for each sequence in the *Emotion Classification* task as Figure 7a. Our analysis reveals that the Expert Choice approach employs 11 out of 16 experts, indicating a lower level of specialization among experts. In contrast, both USMoE and the Token Choice approach use an average of 0.9 to 1 expert per sequence, demonstrating superior expert specialization. Furthermore, we analyze the token dropping behavior of the Expert Choice approach and observe a significant increase in dropping rates when scaling to larger datasets or models, such as pre-training the Transformer-XL Large model on the *One Billion Word* dataset, as shown in Figure 7b. This increase in dropping rates may negatively impact model performance. In contrast, our method maintains a consistently low dropping rate (<0.1), demonstrating its superiority over the Expert Choice approach for scalability. Additionally, our method proves more robust than the Token Choice approach, as it effectively drops irrelevant tokens without compromising performance.

A.4 Implementation Details

For the **Without Training** experiments, we implement our method based on the publicly available MoEE implementation (Li and Zhou, 2024)[1]. Due to resource constraints, we validate our method and the baselines using 4-bit quantization with a batch size of 128. For the *OLMoE-1B-7B* model, we conduct experiments on a single H100 GPU, while for the *Qwen1.5-MoE-A2.7B* and *DeepSeekMoE-16B* models, we utilize two H100 GPUs.

The base Transformer-XL variant (Chen et al., 2023b) comprises four Transformer decoder layers, each with an input dimension of 256. Each layer includes a self-attention mechanism with eight attention heads, followed by a Feed-forward Neural

[1] https://github.com/tianyi-lab/MoE-Embedding

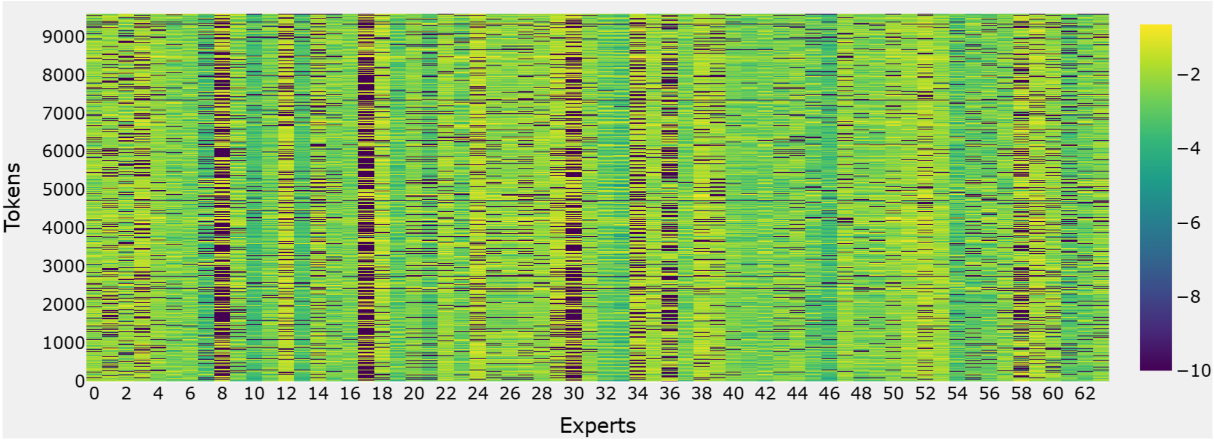
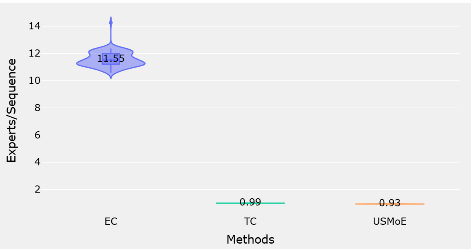
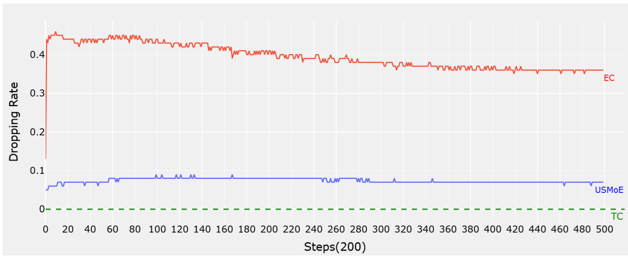


Figure 6: **USMoE Router** visualization for the **OLMoE-1B-7B** model on the *Emotion Classification* task. The scores of selected experts are replaced with *-10.0* (lower than the minimum score) to enhance visualization. Best viewed in color.



(a) Number Experts per Sequence of USMoE, TC, and EC on *Emotion* dataset.



(b) Token Dropping of USMoE, Token Choice (TC), Expert Choice (EC) for Pre-training on *One Billion Word* dataset.

Figure 7: Comparison of the number of experts per sequence for USMoE, Token Choice (TC), and Expert Choice (EC) on the *Emotion* dataset using the **OLMoE-1B-7B** model, along with a comparison of token dropping rates for USMoE, TC, and EC during pre-training on the *One Billion Word* dataset.

Network (FFN) that has an inner dimension of 512. The dropout ratio is set at 0.1. We divide the FFN into 16 experts, each with the same dimensions. For the larger variants, we scale the model up to twelve layers.

Our experiments are based on the publicly available SMoE-Dropout implementation (Chen et al., 2023b)[2]. The pre-training experiments were conducted using a single H100 GPU, while the fine-tuning experiments were performed on a single A100 GPU. It is important to note that parallel training on multiple GPUs may produce different results.

A.4.1 Pre-training Experiments

We provide the USMoE implementation details for pre-training our Transformer-XL base and large on `enwik8`, `text8`, `WikiText-103`, and `One Billion Word` in Table 11.

A.4.2 Fine-tuning Experiments

To perform the fine-tuning experiments, we utilize the same model architecture as in the pre-training phase. Table 12 presents the implementation details for the fine-tuning experiments conducted across four different datasets.

[2] https://github.com/VITA-Group/Random-MoE-as-Dropout

Category	**Model**	**Dataset**	**Setting**	**Router**	**TC**	**EC**	**MoEE**	**USMoE**
Classification	OLMoE-1B-7B	Emotion	None	24.1	24.5	26.3	25.1	**28.2**
			Prompt	27.6	49.9	49.0	44.5	**51.7**
		Toxic	None	51.9	58.9	59.9	51.9	**60.6**
			Prompt	52.3	65.2	61.3	53.4	**66.6**
		Tweet	None	47.7	46.8	48.3	48.4	**52.1**
			Prompt	49.5	58.0	58.4	57.2	**59.2**
Clustering	OLMoE-1B-7B	Medrxiv	None	15.0	17.6	14.8	17.4	**21.2**
			Prompt	15.8	23.9	**27.7**	22.0	26.3
		20Groups	None	12.4	11.8	9.2	11.5	**12.6**
			Prompt	16.7	25.7	26.2	24.4	**34.6**
Pair Classification	OLMoE-1B-7B	SemEval	None	43.6	35.8	31.3	43.6	**45.7**
			Prompt	45.7	46.7	40.9	53.8	**54.2**
		URLCorpus	None	47.0	42.4	39.7	47.8	**54.5**
			Prompt	61.4	77.4	76.9	78.2	**79.5**
Reranking	OLMoE-1B-7B	Ask	None	41.3	41.0	39.0	41.4	**42.4**
			Prompt	43.4	51.9	49.9	50.2	**51.8**
		SciDocs	None	45.5	46.3	46.9	50.8	**59.9**
			Prompt	53.6	69.6	73.1	75.1	**76.6**
		StackOver	None	25.8	24.8	20.1	26.4	**27.0**
			Prompt	28.1	32.5	30.0	34.3	**35.6**
STS	OLMoE-1B-7B	Biosses	None	39.3	13.6	7.7	29.7	**45.4**
			Prompt	51.2	61.8	67.6	70.2	**74.3**
		SickR	None	50.3	46.3	26.4	53.0	**55.1**
			Prompt	51.9	65.7	37.6	66.1	**67.9**
		STS12	None	40.1	8.6	11.1	37.8	**44.6**
			Prompt	51.3	53.8	37.5	63.6	**67.1**
		STS13	None	40.5	21.1	18.2	43.4	**49.8**
			Prompt	52.5	66.5	40.4	72.7	**76.2**
		STS14	None	29.5	13.4	13.3	31.7	**38.7**
			Prompt	41.1	56.8	33.9	64.2	**67.3**
		STS15	None	30.8	27.8	22.5	33.3	**43.3**
			Prompt	46.4	69.3	38.4	66.4	**71.7**
		STS16	None	46.5	38.9	28.9	45.8	**51.7**
			Prompt	52.4	70.1	49.4	68.3	**72.7**
		STSBen	None	42.2	23.4	17.5	44.5	**51.1**
			Prompt	48.6	63.6	48.9	70.7	**71.4**
Summarization	OLMoE-1B-7B	Medrxiv	None	28.4	20.9	21.1	29.8	**30.8**
			Prompt	25.6	28.9	29.7	30.4	**30.9**

Table 8: Performance comparison of USMoE, Token Choice (TC), Expert Choice (EC), and MoEE across MTEB Tasks with *OLMoE-1B-7B* models. The best result for each row is highlighted in **bold**.

Category	**Model**	**Dataset**	**Setting**	**Router**	**TC**	**EC**	**MoEE**	**USMoE**
Classification	Qwen1.5-MoE-A2.7B	Emotion	None	27.2	33.9	30.1	34.3	**35.7**
			Prompt	37.0	48.5	47.4	47.2	**49.9**
		Toxic	None	53.0	61.1	21.3	52.9	**61.4**
			Prompt	53.4	64.5	19.8	54.1	**66.2**
		Tweet	None	51.1	55.9	25.1	55.9	**56.5**
			Prompt	56.1	61.1	38.6	60.7	**61.5**
Clustering	Qwen1.5-MoE-A2.7B	Medrxiv	None	15.3	23.3	21.3	23.0	**23.4**
			Prompt	14.2	24.6	19.8	21.8	**25.1**
		20Groups	None	12.0	**31.5**	25.1	27.4	30.0
			Prompt	14.4	43.8	38.6	38.4	**46.4**
Pair Classification	Qwen1.5-MoE-A2.7B	SemEval	None	42.0	38.8	34.7	42.5	**44.2**
			Prompt	47.0	52.4	46.3	52.4	**54.1**
		URLCorpus	None	49.8	54.9	52.1	60.6	**62.1**
			Prompt	56.7	68.7	65.8	68.2	**72.7**
Reranking	Qwen1.5-MoE-A2.7B	Ask	None	43.1	45.8	43.4	47.3	**47.5**
			Prompt	43.3	48.3	49.1	49.5	**51.4**
		SciDocs	None	49.6	60.6	55.3	**67.0**	65.6
			Prompt	50.9	60.1	55.8	68.7	**73.0**
		StackOver	None	26.2	29.5	26.2	31.1	**31.4**
			Prompt	28.8	31.3	30.2	35.2	**36.6**
STS	Qwen1.5-MoE-A2.7B	Biosses	None	33.8	32.5	34.7	49.6	**52.6**
			Prompt	55.1	55.8	48.5	**68.4**	66.2
		SickR	None	51.0	55.5	40.4	61.0	**63.6**
			Prompt	50.2	59.7	51.1	64.3	**66.3**
		STS12	None	40.2	16.9	18.6	46.3	**48.3**
			Prompt	49.3	25.0	31.8	59.2	**61.4**
		STS13	None	38.1	42.9	44.2	56.7	**61.8**
			Prompt	53.3	57.5	54.6	73.4	**75.7**
		STS14	None	28.1	26.5	25.6	45.4	**48.9**
			Prompt	40.4	38.8	40.7	60.0	**62.7**
		STS15	None	34.8	40.5	38.4	46.1	**48.0**
			Prompt	40.7	52.3	54.2	58.8	**62.5**
		STS16	None	47.6	51.0	48.1	58.1	**59.6**
			Prompt	51.6	64.2	65.1	65.7	**68.2**
		STSBen	None	37.0	37.7	34.7	50.9	**55.5**
			Prompt	45.6	47.8	54.5	64.5	**67.3**
Summarization	Qwen1.5-MoE-A2.7B	Medrxiv	None	28.3	13.4	15.1	**31.2**	29.7
			Prompt	27.0	23.0	21.9	27.3	**48.4**

Table 9: Performance comparison of USMoE, Token Choice (TC), Expert Choice (EC), and MoEE across MTEB Tasks with *Qwen1.5-MoE-A2.7B* models. The best result for each row is highlighted in **bold**.

Category	**Model**	**Dataset**	**Setting**	**Router**	**TC**	**EC**	**MoEE**	**USMoE**
Classification	DeepSeekMoE-16B	Emotion	None	26.1	27.4	26.5	27.6	**27.8**
			Prompt	37.9	48.3	46.1	46.4	**48.8**
		Toxic	None	53.3	**60.4**	58.1	53.1	60.1
			Prompt	53.1	62.4	62.5	53.6	**64.2**
		Tweet	None	51.0	51.9	49.5	**52.6**	52.5
			Prompt	54.9	58.4	57.5	**58.9**	**58.9**
Clustering	DeepSeekMoE-16B	Medrxiv	None	15.1	23.0	17.3	22.0	**25.0**
			Prompt	17.0	25.7	20.9	24.0	**25.8**
		20Groups	None	11.7	13.2	9.7	13.7	**18.7**
			Prompt	18.6	32.3	19.8	33.0	**37.9**
Pair Classification	DeepSeekMoE-16B	SemEval	None	44.6	40.2	32.6	43.5	**46.0**
			Prompt	48.4	47.2	46.6	51.3	**54.6**
		URLCorpus	None	46.4	41.7	41.6	48.6	**57.9**
			Prompt	66.5	72.4	61.1	75.4	**75.9**
Reranking	DeepSeekMoE-16B	Ask	None	41.7	41.1	40.1	42.3	**45.0**
			Prompt	43.5	43.8	44.7	46.9	**49.9**
		SciDocs	None	48.2	50.6	44.7	57.1	**61.9**
			Prompt	58.3	65.6	55.3	72.6	**72.9**
		StackOver	None	25.7	24.9	20.4	27.3	**30.2**
			Prompt	29.7	27.6	22.6	32.3	**33.6**
STS	DeepSeekMoE-16B	Biosses	None	29.5	31.7	27.7	26.8	**35.4**
			Prompt	47.0	40.1	41.5	**57.6**	55.3
		SickR	None	50.4	47.4	29.4	53.1	**56.7**
			Prompt	56.0	61.9	38.7	65.8	**67.7**
		STS12	None	44.0	4.3	13.9	45.0	**46.9**
			Prompt	57.8	31.0	28.4	64.0	**64.2**
		STS13	None	36.0	28.4	27.5	41.1	**49.1**
			Prompt	55.3	56.0	41.2	70.9	**75.2**
		STS14	None	25.4	12.0	13.0	28.2	**37.3**
			Prompt	44.9	41.0	31.1	58.6	**63.8**
		STS15	None	34.8	33.9	25.6	38.7	**40.7**
			Prompt	49.7	46.5	33.0	58.5	**63.0**
		STS16	None	44.9	34.4	33.1	46.9	**55.7**
			Prompt	56.7	58.0	44.2	64.5	**69.7**
		STSBen	None	36.6	18.3	15.8	42.1	**45.6**
			Prompt	54.9	57.7	39.0	67.8	**69.2**
Summarization	DeepSeekMoE-16B	Medrxiv	None	24.9	22.0	18.5	24.4	**28.4**
			Prompt	29.1	24.4	25.7	29.2	**30.9**

Table 10: Performance comparison of USMoE, Token Choice (TC), Expert Choice (EC), and MoEE across MTEB Tasks with *DeepSeekMoE-16B* models. The best result for each row is highlighted in **bold**.

Table 11: Implementation details for pre-training experiments on `enwik8`, `text8`, `WikiText-103`, and `One Billion Word` datasets.

Dataset	Input length	Batch size	Optimizer	Lr	# Iterations
enwik8	512	48	Adam	2.5e-4	100k
text8	512	48	Adam	2.5e-4	100k
WikiText-103	512	22	Adam	2.5e-4	100k
One Billion Word	512	11	Adam	2.5e-4	100k

Table 12: Implementation for fine-tuning experiments on downstream tasks.

Dataset	Input length	Batch size	Optimizer	Lr	# Epochs
SST-2	512	16	Adam	1e-4	15
SST-5	512	16	Adam	1e-4	15
IMDB	512	4	Adam	1e-4	15
BANKING77	512	16	Adam	1e-4	15