A Closer Look into Mixture-of-Experts in Large Language Models

Ka Man Lo *
University of Macau

Zeyu Huang *
University of Edinburgh

Zihan Qiu *
Tsinghua University

Zili Wang
INF Technology

Jie Fu †
Shanghai AI Lab

Abstract

Mixture-of-experts (MoE) is gaining increasing attention due to its unique properties and remarkable performance, especially for language tasks. By sparsely activating a subset of parameters for each token, MoE architecture could increase the model size without sacrificing computational efficiency, achieving a better trade-off between performance and training costs. However, the underlying mechanism of MoE still lacks further exploration, and its modularization degree remains questionable. In this paper, we make an initial attempt to understand the inner workings of MoE-based large language models. Concretely, we comprehensively study the parametric and behavioral features of four popular MoE-based models and reveal some intriguing observations, including 1) Neurons act like fine-grained experts; 2) The router of MoE usually selects experts with larger output norms; 3) The expert diversity increases as the layer increases, while the last layer is an outlier, which is further validated by an initial experiment. Based on the observations, we also provide suggestions for a broad spectrum of MoE practitioners, such as router design and expert allocation. We hope this work could shed light on future research on the MoE framework and other modular architectures. Code is available at `https://github.com/kamanphoebe/Look-into-MoEs`.

1 Introduction

The advent of Large Language Models (LLMs) revolutionized the field of Natural Language Processing. LLM researchers are continually pushing the boundaries of Language Models by scaling up both model size and the column of training data, significantly enhancing the capabilities of these models. This escalation in training cost and complexity necessitates innovative solutions to better balance between pre-training efficiency and model performance. One emerging solution to this end is the Mixture-of-Experts (MoE) (Shazeer et al., 2017) architecture. The MoE framework facilitates the computational efficiency of the model by dynamically routing inputs to a subset of experts, allowing for substantial model scaling while maintaining training costs and leading to numerous influential advancements in the field (Reid et al., 2024; Jiang et al., 2024; Dai et al., 2024; Team, 2024).

Beyond efficiency, another attractive trait of MoE architecture is its modular design and learning paradigm. This modularization allows for flexible and potentially more generalizable handling of diverse data and tasks within a single model by assigning them to specialized experts. Despite its widespread adoption, it remains an open question whether current MoE-based LLMs truly leverage this modularity in knowledge distribution and expert behaviors. In other words, is MoE a simple ensemble of homogeneous experts or a modular combination of heterogeneous experts? Answering this question comprehensively is non-trivial. Therefore, in this paper, we take the first step by investigating four popular MoE-based LLMs (Mixtral 8x7B (Jiang et al., 2024), Mixtral 8x22B, DeepSeekMoE (Dai et al., 2024), and Grok-1[1]) from two critical perspectives: model parameters and model behaviors. We aim to explore common and distinct features and behaviors among different experts, further shedding light on the inner mechanisms of MoE-based models.

Specifically, we examine the correlation between experts' parameters, gates, and their output features given text inputs. Before diving into deeper analyses, we briefly summarize some of our empirical conclusions (detailed in § 6) and observations:

- Neurons in the Feed-Forward Network (FFN)

* Equal contribution.
† Corresponding author.

[1] https://github.com/xai-org/grok-1

Findings of the Association for Computational Linguistics: NAACL 2025, pages 4427–4447
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

layer are fine-grained experts. Both the gate embedding matrix and the expert projection matrix W_{act} perform the choosing operation: the former determines the expert selection while the latter controls the neuron activation. We observe that the similarity heat maps exhibit correlations, suggesting that, from the perspective of W_{act}, the expert neurons can be considered as "tiny" experts, each represented by a single neuron.

- Increasing the number of experts in deeper layers while reducing it in the last layer. This is experimented in Fig. 5. Our observations indicate that the similarities between the parameters and outputs of the experts consistently decrease with increasing layer number, followed by a sudden increase in the last layer.
- Using the norm as the routing mechanism is a reasonable choice. For both Mixtral 8x7B and DeepSeekMoE, we observe that the gate typically selects experts with larger output norms.
- When analyzing the correlation between experts, measuring the similarities between weight matrices is, to some extent, equivalent to assessing the average similarities of expert outputs.
- Training MoE from scratch promotes greater expert diversity than specific initialization schemes. This stems from the observations that stronger correlations (*e.g.*, higher similarities) between parameters and behaviors in Mixtral experts. In contrast, DeepSeekMoE and Grok-1, which are trained from scratch, do not show these correlations.

2 Preliminary: Mixture-of-Experts

Mixture-of-Experts models enhance transformers by replacing the original FFNs with N parallel FFNs combined with a router. These N FFNs are called experts and denoted as E_n for $n \in [1, N]$. The gate $g(\cdot; W_g, k)$, parameterized by W_g and an integer k, assigns the input x to a score distribution over the experts, $g(x; W_g, \mathrm{k}) \in \mathbb{R}^N$. Typically, the gate g consists of a simple linear layer followed by a $\mathrm{softmax}$ and a $\mathrm{Top}\text{-}\mathrm{k}$ function.

Given $x \in \mathbb{R}^{d_{\text{hid}}}$, the output $y \in \mathbb{R}^{d_{\text{hid}}}$ is the weighted sum of the outputs from all experts:

$$y = \sum_{n \in N} g_n(x; W_g, \mathrm{k}) E_n(x)$$

When k for $\mathrm{Top}\text{-}\mathrm{k}$ is smaller than N, only a subset of experts is involved in the computation. This is known as Sparse Mixture-of-Experts (SMoE).

The experts E_n of the models investigated in this paper follow the style in LLaMA (Touvron et al., 2023), which consists of three linear layers and operates as (the subscript n is omitted for brevity):

$$E(x) = W_{\text{down}}(W_{\text{up}}x \odot \sigma(W_{\text{act}}x)) \qquad (1)$$

where $\odot$ represents element-wise multiplication and σ represents the activation function. Given the three projection matrices $W_{\text{up}}, W_{\text{act}} \in \mathbb{R}^{d_{\text{mid}} \times d_{\text{hid}}}$ and $W_{\text{down}} \in \mathbb{R}^{d_{\text{hid}} \times d_{\text{mid}}}$, we define a neuron as the combination of the row vectors $W_{\text{up}}[i, :]$ and $W_{\text{act}}[i, :]$, along with the column vector $W_{\text{down}}[:, i]$. Thus, each expert contains d_{mid} neurons, each with size d_{hid}.

3 Overview

Our experiments are conducted on several open-source MoE models, namely Mixtral 8x7B, Mixtral 8x22B [2], DeepSeekMoE, and Grok-1. We choose these models due to their widespread use and impressive performance across various domains. Additionally, they exhibit complementary characteristics across several key attributes, enabling a robust comparative analysis using control variables. Details are discussed in Append A. To further study the similarities and differences between a standard transformer and a MoE model, we include Mistral 7B (Jiang et al., 2023) as one of our investigated models. Basic information about these models, along with the abbreviations used throughout our paper, is summarized in Tab. 1 and Tab. 4. The analysis is divided into two sections: one focusing on model parameters (**static**) and the other on model behaviors in response to text input (**dynamic**).

Unless otherwise stated (§ 5.1), cosine similarity is employed for all experiments involving similarity measurements. While we acknowledge the existence of other metrics, we primarily use cosine similarity as it is a widely adopted approach (Sun et al., 2024; Zhang et al., 2021).

4 Analysis of Static Parameters

From a high-level perspective, a model's knowledge is encoded in its parameters, making the investigation of weight matrices a natural approach. In this section, we study the correlation between the parameters of: i) MoE experts (and FFNs for

[2] For the Mixtral 8x22B model, we only conduct most of the analyses mentioned in the main context (excluding those in the appendix) due to time limit.

Model	Abbreviation	# MoE layers	# experts	Top-k	Hidden size (d_{hid})	Intermediate size (d_{mid})
Mixtral 8x7B	Mixtral	32	8	2	4096	14336
Mixtral 8x22B	Mixtral-22	56	8	2	6144	16384
Mistral 7B	Mistral	32	N/A	N/A	4096	14336
DeepSeekMoE	DeepSeek	27	64 routed + 2 shared	6	2048	1408
Grok-1	Grok	64	8	2	6144	32768

Table 1: Basic information of models used for analysis. The abbreviations are used throughout our paper.

Mistral), and ii) gate embeddings; which are two vital components of the MoE architecture.

4.1 Weight Matrices of Experts

MoE models replace FFNs in standard transformers with experts. Following Geva et al. (2020); Qiu et al. (2024b), the projection matrices of the experts can be regarded as keys and values: the column vectors of W_{down} represent potential outputs; the row vectors of W_{up} produce weights for each possible output; the row vectors of W_{act} determine whether to activate the corresponding neurons. Thus, examining the weight matrices provides a straightforward way to understand the expert behaviors. We analyze both the matrix and neuron levels to gain insights from different perspectives.

4.1.1 Matrix-level

In this part, we explore the similarity of the three projection matrices W_{up}, W_{act}, and W_{down} among all experts in each layer. The similarity is calculated based on the flattened matrices and is illustrated in Fig. 1. We denote “F” as the Mistral FFN and “SE” as the DeepSeek shared expert. *Note that the figures for different models do not share the same color bar.*

Common [3]. The heat maps of the three matrices exhibit similar patterns. Directly flattening the large weight matrices leads to high-dimensional vectors, so we use principal components analysis (PCA) to reduce these vectors to two-dimensional space. The resulting figures also show that, for Mixtral and DeepSeek, the expert distribution across the three weight matrices is generally comparable. Details on the PCA results are presented in Append C.1.

Mixtrals and Mistral. The cosine similarities between Mixtral experts (S_{ee}) primarily range from 0.2 to 0.4, while the similarities between the experts and the Mistral FFN (S_{ef}) are about 0.6. Yet the values tend to be lower in the deeper layers (22nd-30th for Mixtral and 35th-50th for Mixtral-22). A “dark cross” can be observed in some layers and corresponds to outliers in the 2D space projected by PCA, indicating that the associated expert is relatively distinct from the others. Interestingly, this cross appears most frequently in Expert 3 for Mixtral, suggesting that this expert may have learned some unique attributes. It is noteworthy that the cross usually extends across the entire heat map, including the last row of the FFN. Thus, when an Mixtral expert differs from other experts, it is also less similar to the Mistral FFN.

DeepSeek and Grok. The shared experts of DeepSeek are implemented as a single MLP block with a larger hidden size than the routed experts, preventing direct comparison of their flattened vectors; thus, we omit them from this experiment. Fig. 1 demonstrates that the similarities between the DeepSeek routed experts and Grok experts are close to zero. While Mixtrals' training method remains unrevealed, it is known that DeepSeek and Grok are trained from scratch. This suggests that Mixtrals may have been trained using special schemes, resulting in less diverse experts compared to those trained from scratch (Wu et al., 2022).

4.1.2 Neuron-level

In § 4.1, we measure the parameter similarity between experts at the matrix level. However, the calculation of cosine similarity is position-dependent. If the neurons of two experts are similar but in different orders, the similarity of their weight matrices will be significantly lower than expected. To address this, we propose two approaches to investigate the correlation at the neuron level: averaging and reordering. Averaging simply averages the rows (for W_{up} and W_{act}) or the columns (for W_{down}) of the weight matrices, and then calculates the cosine similarity of the resulting vectors across experts. For reordering, we apply the Jonker-Volgenant algorithm (Jonker and Volgenant, 1988), which is typically used for solving linear assignment problems, to find the optimal order of neurons so that the cosine similarity between two experts is maximized.

We describe the results of the reordering method

[3] The observations shared by *all* of our investigated models are written in the Common part.

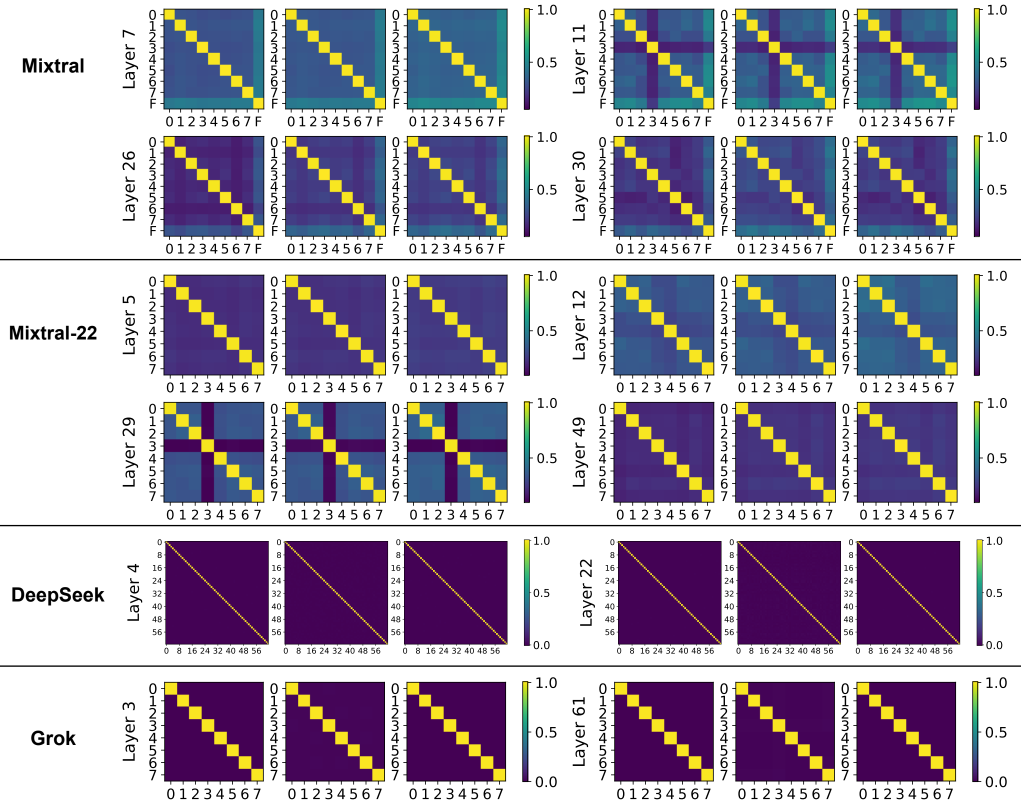

Figure 1: Matrix-level similarity heat maps of expert weight matrices. Each layer contains three heat maps, corresponding to W_{up}, W_{act}, and W_{down}, respectively. The tick numbers refer to the expert indices. "F" denotes the Mistral FFN.

below and provide the details of the averaging approach in Append D. Additionally, the projection of neurons into low-dimensional spaces using PCA can be found in Append C.2. Due to the heavy computation, we only select several layers for the reordering calculation. Note that the matrices are reordered separately. We measure Kendall's τ coefficient between the index sequences before and after reordering, whose value increases when the two sequences exist strong agreement. Tab. 2 depicts the common similarity growth after reordering and the average Kendall's coefficient $\bar{\tau}$ over the selected layers. The order of Mixtral neurons changes little (resulting in a large τ), and hence nearly unchanged similarities. Despite the substantial similarity increase for DeepSeek and Grok after reordering, their overall values remain around 1e-2.

Model	Order of Growth	$\bar{\tau}$
Mixtral	1e-3	0.75
DeepSeek	100	-0.0002
Grok	100	-0.0003

Table 2: Reordering results of expert neurons.

4.2 Gate Embedding

The gate embedding of our investigated MoE models is implemented as a linear layer W_g with size $\mathbb{R}^N \times \mathbb{R}^{d_{\text{hid}}}$, where N is the number of experts. The gate serves as a crucial component of MoE, making it essential to study its attributes to understand MoE functionality better. In addition, since each row vector in the gate embedding determines expert selection, some correspondence may exist between W_g and the expert weights.

To investigate this, we measure the similarities between the gate embedding vectors $W_g[n,:]$ for $n \in [1, N]$. For computational simplicity, we compare them with the neuron-level averaging (instead of the reordering) heat maps of experts presented in Append D, with qualitative analyses detailed in Append E. Specifically, we found that, for all four MoE models, the patterns in the heat maps of v and of expert neurons $W_{\text{act}}[i,:]$ are partially alike in some layers (*i.e.*, the same coordinates in both heat maps exhibit relatively higher or lower values simultaneously).

Therefore, we further conduct a quantitative analysis of their similarity values. In particular, we perform linear regression on the paired similarity dataset (X, Y), where X denotes the similarities of $W_g[n, :]$, and Y denotes the neuron-level similarities of W_{up}, W_{act}, or W_{down}. Tab. 3 describes the average square of Pearson correlation coefficients over all layers (R^2_{avg}), while Tab. 5 lists the Pearson correlation coefficient (R) for each layer. As shown in Tab. 3, the correlation between the similarities of the gate vectors and those of W_{act} is significantly stronger than that with W_{up} and W_{down}. For the (X, Y_{act}) pair, although Mixtral and DeepSeek have similar R^2_{avg} values, the R^2 of Mixtrals fluctuate between 0.1 and 0.7, while the R^2 of DeepSeek remains close to 0.4. Furthermore, we can see from Tab. 5 that (X, Y_{act}) for both Mixtral and DeepSeek show positive correlations, whereas (X, Y_{act}) for Grok turn to negative correlations starting from the intermediate (after 25th) layers. We note that the function of W_g and W_{act} is analogous: the former determines expert selection while the latter is responsible for choosing which neurons to activate. Therefore, they may learn similar knowledge to effectively perform the *choosing* operation, which explains the observed correlation.

Model	(X, Y_{up})	(X, Y_{act})	(X, Y_{down})
Mixtral	0.06	0.33	0.07
Mixtral-22	0.13	0.26	0.13
DeepSeek	0.00	0.40	0.00
Grok	0.04	0.15	0.04

Table 3: Average square of Pearson correlation coefficients over all layers (R^2_{avg}) for three paired dataset.

4.3 Summary

Here, we conclude the key observations from the analysis of static parameters: **i)** Mixtral might contain expert(s) with unique attributes, as evidenced by the frequent presence of dark crosses in Fig. 1. **ii)** The similarities of DeepSeek and Grok expert weight matrices are generally lower than those in Mixtrals. As mentioned in § 4.1.1, the matrix-level similarities of DeepSeek and Grok experts are typically close to zero, whereas Mixtrals' expert similarities average around 0.3. **iii)** The weights of different experts become less similar in deeper layers, as observed in the Mixtrals' heat maps in Fig. 1. **iv)** W_{up}, W_{down}, and W_{act}, share similar patterns in their similarity heat maps (Fig. 1). **v)** The similarities of W_g and of W_{act} show either positive or negative association. Tab. 3 depicts the R^2_{avg} values, where the pairing of W_g and W_{act} achieves the highest correlation across all four models.

5 Analysis of Dynamic Behaviours

The previous experiments examine the MoE models via their parameters, without involving any input. In this section, we feed text sequences into the MoE models to further study their actual behaviours given various inputs. Specifically, we analyze the outputs of the experts and gates.

To this end, two stages are required for inference. In the first stage, we simply pass the input x through the network using the original Top-k setting and store the output z_i of every layer i. In the second stage, we iterate through the layers. During the i-th iteration, we feed z_{i-1} into the i-th layer (for the first layer, x is employed as the input), set Top-k $=$ ALL, and record the outputs from all the experts in the i-th layer. Note that each layer has its own individual forward pass in the second stage. Intuitively, our goal is to examine the experts' behaviors when provided with the original inputs.

Input data. We utilize a *short input* and a *long input* for the experiments in this section. For the short input, we employ the first few words of the input from another MoE-related work (Cai et al., 2024) [4]. For the long input, we adopt 10 sequences from the test set of the WikiText-103 (Merity et al., 2016) dataset, totaling approximately 1100 tokens. The sequences in WikiText-103 cover a variety of domains, with the 10 sequences we used spanning topics such as music, design, and construction. To ensure the robustness of our findings, we repeat experiments requiring the long input (§ 5.1, § 5.2) using additional datasets with over 80K tokens, including GSM8K (Cobbe et al., 2021) and Magicoder- Evol-Instruct-110K (Wei et al., 2024b). See Append F for details. *The observations of these additional, subject-specific datasets align with the results described in the main context, demonstrating the universality of our conclusions.*

We also conduct experiments for analyzing intermediate states of experts and routing patterns. Due to the page limit, these experiments are presented in Append H and Append I, respectively.

5.1 Outputs of Experts

Since experts are ideally learned to specialize in different aspects, it is natural to question the similarities and differences between the outputs of selected and non-selected experts. In this experiment, we

[4] The specific tokens are <*s*>, *As*, *an*, *open*, *source*, *alternative*, *to*, where the start of the sentence symbol <*s*> does not applicable for the Grok tokenizer.

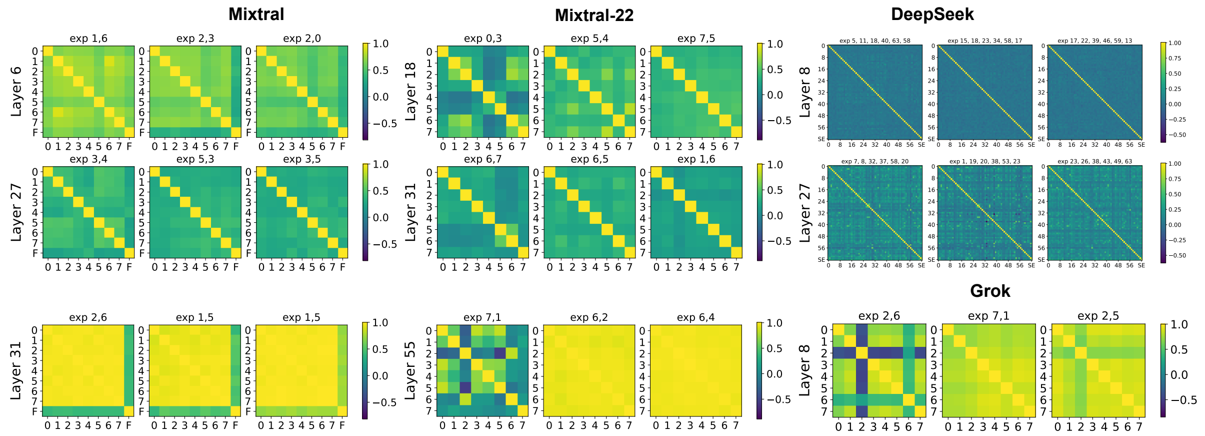

Figure 2: Similarity heat maps of expert output features using the short input. The top k experts for each token are shown on top of each heat map. The tick numbers refer to the expert indices. "F" and "SE" denote the Mistral FFN and the DeepSeek shared expert, respectively.

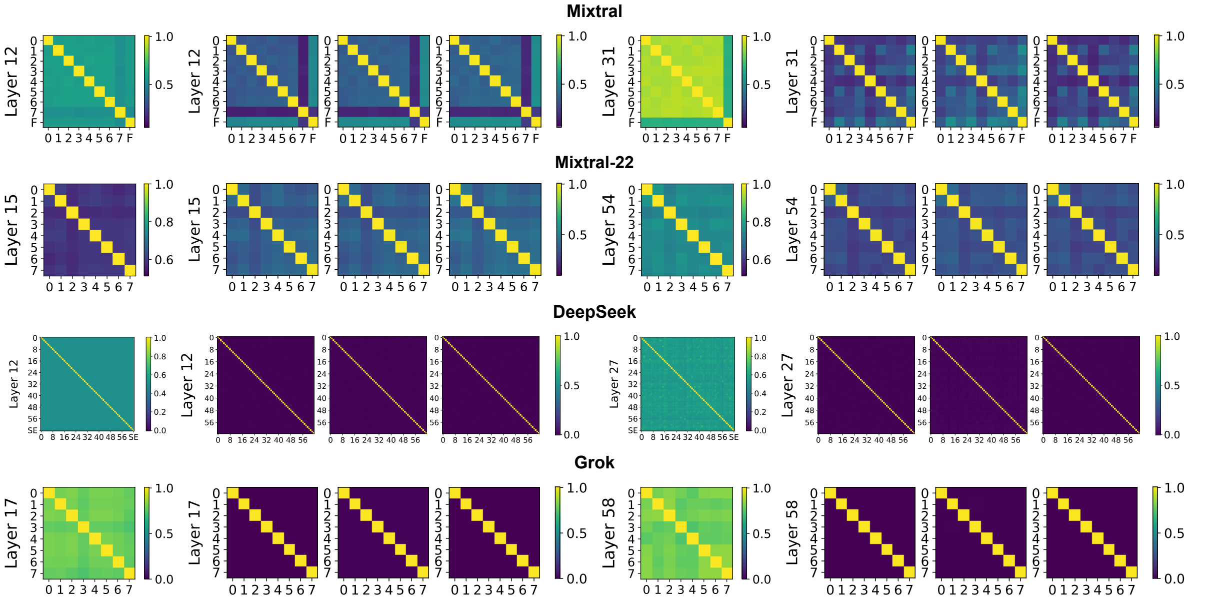

Figure 3: Average similarity heat maps of expert output features using the long input, plotted along with the matrix-level similarity heat maps. The tick numbers refer to the expert indices. "F" denotes the Mistral FFN.

measure the correlation between the output feature vectors of experts. We plot the similarity heat maps for three tokens in the short input (Fig. 2) and the average heat map across all tokens in the long input (Fig. 3). For the long input, we use *angular similarity* instead of cosine similarity for measurement, as the similarities need to be averaged, ensuring that the values range from 0 to 1:

$$\text{angular\_sim} = 1 - \frac{\arccos\left(\text{cosine\_sim}\right)}{\pi}. \quad (2)$$

For clarity, the average similarity heat maps are plotted alongside the matrix-level similarity graphs of the expert weight matrices. Fig. 9 further depicts the results from additional datasets, which are consistent with those of the long input.

Mixtrals and Mistral. The graphs for the short input indicate that the outputs from chosen experts tend to be more similar, possibly due to their generally larger norms, which we will discuss in § 5.2. Overall similarities are relatively low in the deeper (22^{nd}-27^{th} for Mixtral and 30^{nd}-50^{th} for Mixtral-22) layers, whereas many values exceed 0.8 in the last few layers. Furthermore, dark crosses often appear in the graphs, with the experts corresponding to these dark crosses often being more similar to the Mistral FFN (*i.e.*, bright color in the last row). For the long input, the average heat maps show patterns akin to neuron-level similarity graphs, including the presence of dark crosses. The similarities also decrease with increasing layer depth, except in the last layer. In addition, we have $S_{\text{ee}} > S_{\text{ef}}$ for both inputs. Most of these observations align with the previous analyses of static parameters (§ 4.3), implying that measuring the similarity of weights, in some aspects, is equivalent to measuring the average similarity of outputs.

DeepSeek. Given the short input, most similarities are around zero, while the values in the last layer are significantly larger. Again, the similarities between experts chosen by the gate are likely to be higher, although this difference occurs much less frequently than in Mixtrals. The average similarities for the long input also approach zero. Moreover, the number of "small rectangular" with relatively light color in the graphs decreases as the layer depth increases (except for the last layer), meaning that the average similarities gradually decline.

Grok. Surprisingly, the similarities between the output features remain high for all tokens in the short input, indicating the experts exhibit similar behaviours. However, the similarities of their weight matrices are mostly zeros (§ 4.1.1). We speculate that this may be due to the relatively large size of each Grok expert, allowing each to learn comprehensive knowledge and behave similarly despite having distinct parameters. When averaging the similarities for the long input, some of the resulting average heat maps display patterns similar to those of the W_{act} figures. This relationship aligns with the observations made for Mixtrals.

5.2 Norms of Expert Outputs and Gate Scores

In § 5.1, we find that the outputs from chosen experts tend to be more alike. To investigate the possible reasons for this observation, we employ the short input to study the relationship between the experts' L2 norm and the gate decision in this experiment. The calculated norms, along with the gate scores, are plotted in Fig. 4. In Append G, we repeat this experiment using the long input and additional datasets, and the results also support the "higher norm, higher score" observation.

Mixtrals. We found that the two experts chosen by the gate usually output feature vectors with the highest norms, which reveals that the norm might be one of the key factors in gate decisions. This finding agrees with the router's design in CompeteSMoE (Pham et al., 2024), which selects experts based on their output norms. It also helps explain why the outputs of the chosen Mixtrals and DeepSeek experts tend to be more alike (§ 5.1). In Fig. 4, we observe that the gate scores assigned to the top-1 experts are usually much higher than those of the others, including the second place. This demonstrates that the gate is learned to strengthen the confidence of its decision during training. On the other hand, the deeper the layer, the larger the norm, which is similar to the growth in standard models (Shleifer et al., 2021).

DeepSeek. In contrast to the observation about Mixtrals' experts, the gate decision appears to depend less obviously on the output norms of DeepSeek experts. However, the top-1 experts often score much higher than the remaining candidates. The magnitude of the norms increases with depth, although the increment is less pronounced than in Mixtrals. In the last layer, the variance of norms becomes greater.

Grok. While the scores of the top-1 experts are higher than those of the others, no correspondence between the norms and the gate scores is observed. One possible reason could be the relatively low activation ratios of GeLU (see Append H), which may lead to a weaker dependence on the norm for gate decisions. Besides, unlike Mixtrals and DeepSeek, the magnitude of the norms hardly changes across depth, and some of the norm values can be less than 1, which is rare in the other two models.

5.3 Summary

The observations of dynamic behaviours are concluded below: **i)** The outputs of Mixtrals and DeepSeek experts in deep (last) layers are less (much) alike. This can be seen in the heat maps for both the short (Fig. 2) and long (Fig. 3) inputs. **ii)** The average heat maps of expert outputs resemble the neuron-level similarity graphs (Fig. 3), implying that weight similarity measurements can reflect output similarity. **iii)** Grok experts exhibit high output similarity (Fig.2), likely due to their larger sizes. **iv)** For Mixtrals and DeepSeek, experts generating feature vectors with larger norms tend to receive higher gate scores, as shown in Fig. 4. We further verified this observation in Fig. 10.

6 Discussion

Based on our analyses, we offer several suggestions for MoE models across various aspects.

Neuron-level experts. Intuitively, the gate embedding matrix W_g determines expert selection while W_{act} is responsible for choosing which neurons to activate. Meanwhile, we find that the similarities of

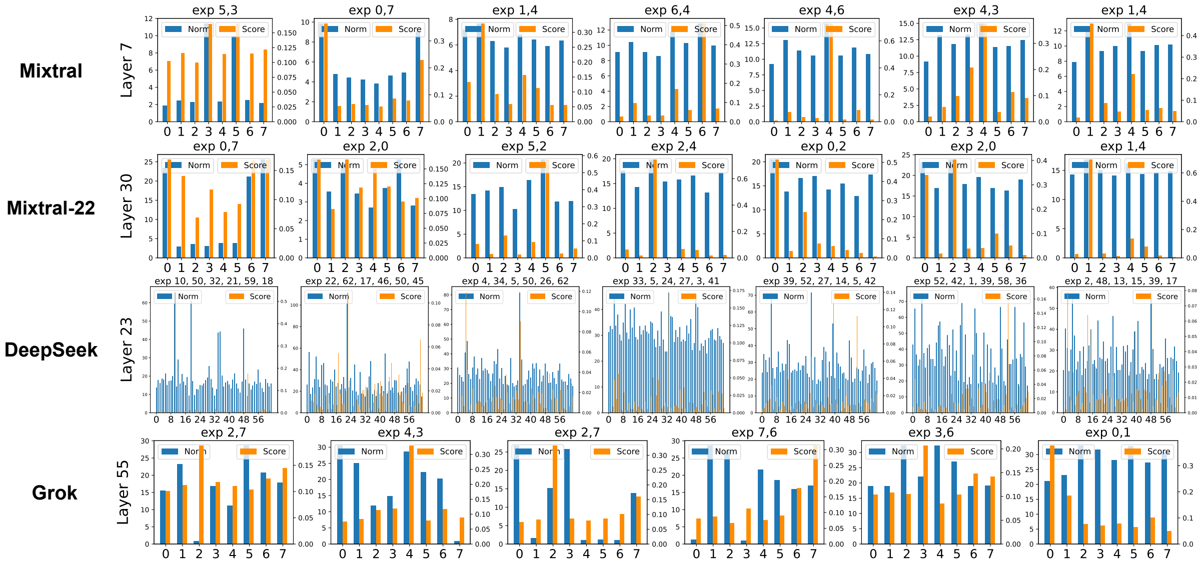

Figure 4: The experts' L2 norms and the gate scores of the short input. Each token's k experts are shown on top of each heat map. Each number in the horizontal axis refers to an expert index.

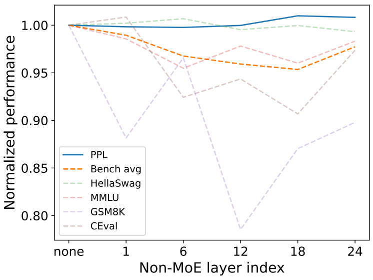

Figure 5: Normalized model performance across benchmarks for the dynamic expert numbers experiment. Solid line represents lower is better, while dashed line represents higher is better. "Bench avg" refers to the average performance over the four benchmarks evaluated.

W_g and of W_{act} show association. This implies that neurons may function as more fine-grained experts. Therefore, operations on experts, such as division, construction, and composition, should be further studied at the micro level. For instance, MoEfication (Zhang et al., 2021) and EMoE (Qiu et al., 2024a) construct MoE experts by splitting the MLP layers of a dense model, suggesting our findings from a similar perspective.

Model architecture. Given that the similarities between experts tend to be relatively low (high) in deep (last) layers, one can consider increasing the number of experts in the deeper layers while reducing it in the last layers. In addition, since the gate frequently selects experts with larger output norms, employing norm-based routing mechanism is a reasonable approach. Empirical evidence from Pham et al. (2024) supports this effectiveness.

We conduct an initial experiment to provide practical experience into our suggestion regarding dynamic expert numbers across layers. Specifically, we train six MoE models from scratch, each containing 24 layers and 3.6B total parameters, using approximately 120B tokens. One of the six models is composed of 24 MoE layers, while the others comprise only 23 MoE layers, with one conventional non-MoE layer positioned at different indices. Details of the model architecture are provided in Tab.7. As displayed in Fig.5 and Tab.6, the average model performance (*i.e.*, PPL and Bench avg) gradually degrades as the non-MoE layer index increases, whereas a slight improvement appears when the non-MoE layer is placed at the last position (24th). This highlights the growing importance of multiple expert networks in deeper layers, excluding the last one, which aligns with our observations and suggestions.

Correlation measurement. Analyzing expert correlations through weight matrix similarities yields partially equivalent results to those from output feature vector similarities across considerable tokens. Thus, assessing weight matrices offers a broader overview, while examining individual token outputs allows for more detailed analysis.

Training scheme. The training method for Mixtral has not been publicly announced. However, we observed certain characteristicss shared by Mixtral experts (*e.g.*, relatively high similarities of weight matrices), and a notable relationship between these experts and the Mistral FFN (*e.g.*, similar intermediate states in Fig. 12). Consequently, we conjecture that the Mixtral model may be trained using special initialization schemes other than from scratch, *e.g.*, upcycling (Komatsuzaki et al., 2022) from Mistral, that is, copying all experts from the FFN. On the contrary, the experts of DeepSeek and Grok, which are known to be trained from scratch, show weaker correlations than Mixtral experts in our experiments. Similarly, Wei et al. (2024a) tracks changes in expert similarities throughout the training process, observing that upcycled experts exhibit greater similarity compared to those randomly initialized. Hence, we speculate that training a MoE model from scratch shows stronger potential to facilitate the diversification of experts compared with certain initialization approaches.

7 Related Work

Due to the page limit, we focus on existing works analyzing MoEs. An extended related work section for MoE LLMs can be found in Append B.

Most existing works analyze MoE from the router's perspective by observing expert selections. Early works have observed the unstable choices in the router (Zuo et al., 2021; Chi et al., 2022; Dai et al., 2022). More recent studies find the standard routers do not show clear specialization at the domain level (Jiang et al., 2024; Dai et al., 2024) and primarily route based on token ID instead of high-level semantics (Xue et al., 2024). Shi et al. (2024) shows that Top-2 and Rank-k routing result in different model behaviours and proposes a new self-contrast decoding method to determine the next-token distribution based on this finding.

Other works investigate the expert's similarity (Wu et al., 2022), uncovering and utilizing redundancies among experts for efficient inference (Li et al., 2023; Lu et al., 2024). Zhang et al. (2024) reveals the redundancy within experts and perform pruning based on their similarities. Liu et al. (2023); Qiu et al. (2023) notice the connection between routing connection and expert computation, and utilize the average of the experts' first-layer weights to guide routing. Pham et al. (2024) proposes adding the expert's output norm as a supervision signal for routing training. Chen et al. (2022) empirically and theoretically proves that a two-layer MoE CNN is able to learn cluster-center features via specializing experts to specific portions of the data. While these works provide insights into MoE from one or two viewpoints, our work offers a systematic analysis and comparison focusing on transformer-based MoE LLMs.

As mentioned in previous sections, several existing works share some relevance to our findings, and thus can be seen as supportive. However, their proposed ideas and methods are different from ours. For instance, rather than revealing the nature of the preference for large output norms in (conventional top-k) routing, as we analyze, CompeteSMoE (Pham et al., 2024) designs a norm-based router to introduce this tendency manually; MoEfication (Zhang et al., 2021) splits MLP layers of a dense model to construct MoE experts, while our study highlights that the neurons of an expert can be seen as tiny experts. Moreover, many of our observations are novel, such as the correlation between the router embedding matrix and the expert weight matrix, as well as the equivalence between parameter and output measurement for experts. Therefore, we believe that our work offers valuable insights into MoE LLMs for the community.

8 Conclusion

In this paper, we initially attempt to investigate the inner working mechanisms of MoEs by studying the parameters and outputs of four different MoE models. We summarize our empirical observations and propose practical suggestions across various aspects. While it is premature to conclude whether MoEs genuinely learn heterogeneous experts, some of our experiments indicate that specific architectural designs (*e.g.*, the number of experts) and training frameworks may facilitate expert specialization. We hope this work can provide inspiring insights and serve as a valuable foundation for future research on MoE and other modular architectures.

9 Limitations

The limitations of our work include: 1) Although the models we investigated cover several common designs of MoE, our analysis does not encompass all aspects (e.g., other routing strategies like top-1 routing or model architectures that place MoE layers at every other layer); 2) Despite the availability of other metrics, we primarily adopt cosine similarity in our experiments involving similarity measurement, as it is a widely used approach (Pham et al., 2024; Chen et al., 2022); 3) We mainly focus on the pretrained base model but seldom explore the behaviours of models after fine-tuning. Analyzing the changes in expert behaviours during the fine-tuning process could yield valuable insights.

References

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. 2024. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*.

Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. 2022. Towards understanding the mixture-of-experts layer in deep learning. *Advances in neural information processing systems*, 35:23049–23062.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, et al. 2022. On the representation collapse of sparse mixture of experts. *Advances in Neural Information Processing Systems*, 35:34600–34613.

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

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.

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

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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*.

Roy Jonker and Ton Volgenant. 1988. A shortest augmenting path algorithm for dense and sparse linear assignment problems. In *DGOR/NSOR: Papers of the 16th Annual Meeting of DGOR in Cooperation with NSOR/Vorträge der 16. Jahrestagung der DGOR zusammen mit der NSOR*, pages 622–622. Springer.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. 2022. Sparse upcycling: Training mixture-of-experts from dense checkpoints. *arXiv preprint arXiv:2212.05055*.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. 2023. Merge, then compress: Demystify efficient smoe with hints from its routing policy. *arXiv preprint arXiv:2310.01334*.

Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J Reddi, Ke Ye, Felix Chern, Felix Yu, Ruiqi Guo, et al. 2022. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. *arXiv preprint arXiv:2210.06313*.

Zeyu Leo Liu, Tim Dettmers, Xi Victoria Lin, Veselin Stoyanov, and Xian Li. 2023. Towards a unified view of sparse feed-forward network in pretraining large language model. *arXiv preprint arXiv:2305.13999*.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. 2024. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. *arXiv preprint arXiv:2402.14800*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

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

Zihan Qiu, Zeyu Huang, and Jie Fu. 2023. Emergent mixture-of-experts: Can dense pre-trained transformers benefit from emergent modular structures? *arXiv preprint arXiv:2310.10908*.

Zihan Qiu, Zeyu Huang, and Jie Fu. 2024a. Unlocking emergent modularity in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2638–2660.

Zihan Qiu, Zeyu Huang, Youcheng Huang, and Jie Fu. 2024b. Empirical study on updating key-value memories in transformer feed-forward layers. *arXiv preprint arXiv:2402.12233*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

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. *arXiv preprint arXiv:1701.06538*.

Yikang Shen, Zhen Guo, Tianle Cai, and Zengyi Qin. 2024. Jetmoe: Reaching llama2 performance with 0.1 m dollars. *arXiv preprint arXiv:2404.07413*.

Yikang Shen, Zheyu Zhang, Tianyou Cao, Shawn Tan, Zhenfang Chen, and Chuang Gan. 2023. Moduleformer: Learning modular large language models from uncurated data. *arXiv preprint arXiv:2306.04640*.

Chufan Shi, Cheng Yang, Xinyu Zhu, Jiahao Wang, Taiqiang Wu, Siheng Li, Deng Cai, Yujiu Yang, and Yu Meng. 2024. Unchosen experts can contribute too: Unleashing moe models' power by self-contrast. *arXiv preprint arXiv:2405.14507*.

Sam Shleifer, Jason Weston, and Myle Ott. 2021. Normformer: Improved transformer pretraining with extra normalization. *arXiv preprint arXiv:2110.09456*.

Chenyang Song, Xu Han, Zhengyan Zhang, Shengding Hu, Xiyu Shi, Kuai Li, Chen Chen, Zhiyuan Liu, Guangli Li, Tao Yang, et al. 2024a. Prosparse: Introducing and enhancing intrinsic activation sparsity within large language models. *arXiv preprint arXiv:2402.13516*.

Yixin Song, Haotong Xie, Zhengyan Zhang, Bo Wen, Li Ma, Zeyu Mi, and Haibo Chen. 2024b. Turbo sparse: Achieving llm sota performance with minimal activated parameters. *arXiv preprint arXiv:2406.05955*.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. 2024. Transformer layers as painters. *arXiv preprint arXiv:2407.09298*.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Tianwen Wei, Bo Zhu, Liang Zhao, Cheng Cheng, Biye Li, Weiwei Lü, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Liang Zeng, et al. 2024a. Skywork-moe: A deep dive into training techniques for mixture-of-experts language models. *arXiv preprint arXiv:2406.06563*.

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

Lemeng Wu, Mengchen Liu, Yinpeng Chen, Dongdong Chen, Xiyang Dai, and Lu Yuan. 2022. Residual mixture of experts. *arXiv preprint arXiv:2204.09636*.

Shaohua Wu, Jiangang Luo, Xi Chen, Lingjun Li, Xudong Zhao, Tong Yu, Chao Wang, Yue Wang, Fei Wang, Weixu Qiao, et al. 2024. Yuan 2.0-m32: Mixture of experts with attention router. *arXiv preprint arXiv:2405.17976*.

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*.

Xiaofeng Zhang, Yikang Shen, Zeyu Huang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2022. Mixture of attention heads: Selecting attention heads per token. *arXiv preprint arXiv:2210.05144*.

Zeliang Zhang, Xiaodong Liu, Hao Cheng, Chenliang Xu, and Jianfeng Gao. 2024. Diversifying the expert knowledge for task-agnostic pruning in sparse mixture-of-experts. *arXiv preprint arXiv:2407.09590*.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2021. Moefication: Transformer feed-forward layers are mixtures of experts. *arXiv preprint arXiv:2110.01786*.

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*.

Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Tuo Zhao, and Jianfeng Gao. 2021. Taming sparsely activated transformer with stochastic experts. *arXiv preprint arXiv:2110.04260*.

Appendix

A Model Selection

Our experiments are conducted on Mixtral 8x7B, DeepSeekMoE, and Grok-1. We choose these models due to their widespread use and impressive performance across various domains. Additionally, these models are complementary in several crucial attributes, such as training scheme, activation functions, top-k settings, and the number of experts, as listed in Tab 1 and Tab 4. This allows for a comparative analysis with controlled variables and encompasses a wide range of parameter sizes, from rather small (16B) to relatively huge (314B). Hence, we believe that the findings derived from these four models are fairly robust, despite the limited number of models examined.

B Extended Related Work

MoE LLMs. MoEs have garnered significant attention in recent years due to their ability to efficiently scale model capacity with minimal computational overhead. Most current transformer-based MoE LLMs adopt a typical architecture design that replaces the original FFN with multiple expert networks and a sparse gating network (Wei et al., 2024a; Wu et al., 2024; Dai et al., 2024; Xue et al., 2024; Jiang et al., 2024; Zoph et al., 2022). JetMoE (Shen et al., 2024) and ModuleFormer (Shen et al., 2023) incorporate Mixture of Attention Heads (Zhang et al., 2022) into their model, achieving further sparsity. A recent survey (Cai et al., 2024) provides a comprehensive review of both the algorithmic and system design aspects of MoEs. For this study, we select four representative candidates among current open-sourced MoE LLMs for analysis to gain intriguing insights.

C Projection of Expert Matrices in Low-dimensional Space

C.1 Matrix-level

To better understand the relationships among experts, we employ principal components analysis (PCA) to project the flattened vectors of weight matrices into two-dimensional space. The vectors are standardized before applying PCA. Fig. 6 depicts the resulting 2D projection.

Mixtral and Mistral. Consistent with the observations in § 4.1.1, the figures for the three matrices appear similar. Generally, about half of the Mixtral experts cluster closely together and near the Mistral FFN, while the others locate much farther away. Moreover, the outliers correspond to the dark crosses.

DeepSeek. Only routed experts are considered due to differences in hidden sizes. Because several outliers exist, causing the remaining data points to be densely gathered, we remove them using the DBSCAN algorithm with $\epsilon = 50$ and plot the rest in Fig. 6. It can be observed that the experts distribute rather densely, especially for W_{up}. Although the distribution of experts varies for three matrices, the figures for W_{up} and W_{down} are more similar than those of the gate matrix.

Grok. Typically, about half of the Grok experts densely gather for W_{up} and W_{down}. The other half turns out to be outliers even though no dark cross were observed before. . Furthermore, the outliers of the three matrices partially coincide.

C.2 Neuron-level

To project the neurons into a 2D or 3D space, each row vector of W_{up} and W_{act}, or each column vector of W_{down}, is treated as a single data point. Standardization is then applied, following by PCA. The visualization of the principal components is illustrated in Fig. 7. Different colors refer to neurons belonging to different experts.

Common. The vast majority of neurons gather in the low-dimensional space. In some layers, the distribution of neurons forms a special shape, such as a cross or a thick line, which appears the most often for W_{down}, followed by W_{up}, and finally W_{act}. Compared to ellipses, these shapes indicate that the neurons are relatively more similar.

Mixtral and Mistral. The neurons in the Mistral FFN distribute more densely than those of the Mixtral experts. Notably, the distribution shape of neurons in the FFN and experts are usually alike, even for the outliers.

DeepSeek and Grok. The number of outliers is a bit greater tahn that observed in Mixtral.

D Averaging Expert Neurons

To investigate expert correlation at the neuron level, the averaging approach simply averages the rows (for W_{up} and W_{act}) or the columns (for W_{down}) of the weight matrices and then calculates the similarity of the resulting vectors across experts. Fig. 8 displays the graphs.

Common. The heat maps of W_{up} and W_{down} are nearly identical to those presented in § 4.1.1. Yet

Model	Training scheme	Activation	# Total layers	# Total params	# Activated params
Mixtral	unknown (upcycling)	SiLU	32	46.7B	12.9B
Mixtral-22	unknown (upcycling)	SiLU	56	141B	39B
Mistral	from scratch	SiLU	32	7.3B	7.3B
DeepSeek	from scratch	SiLU	28	16.4B	0.3B
Grok	from scratch	GeLU	64	314B	78.5B

Table 4: Additional information of chosen models.

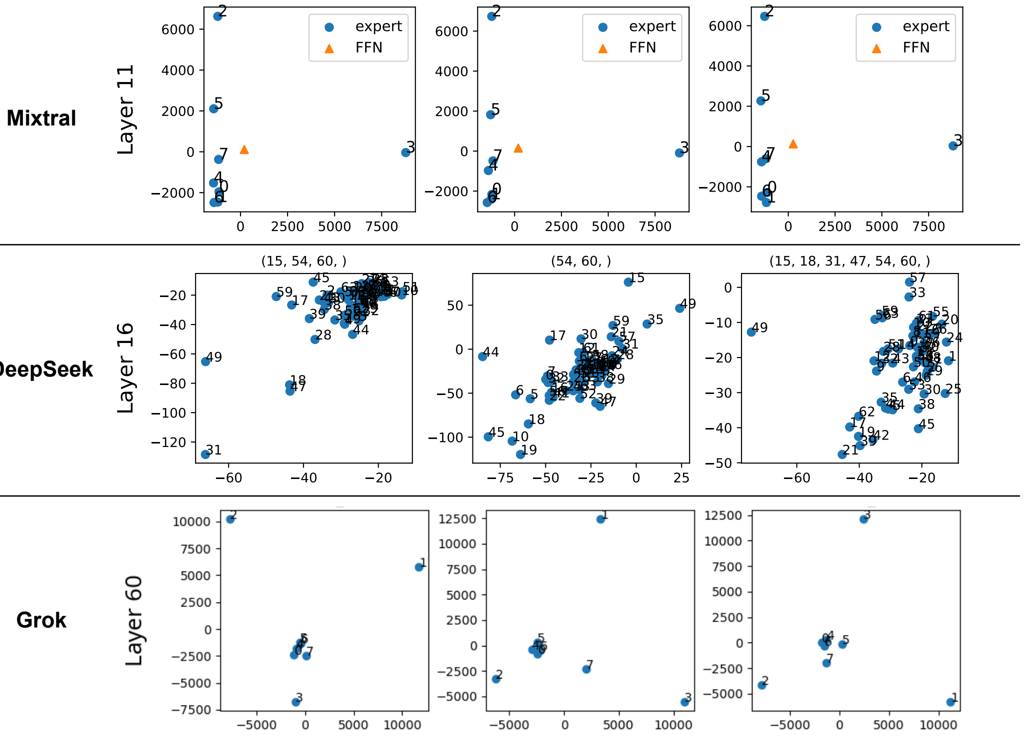

Figure 6: Projection of expert matrices in 2D space. Each layer contains three graphs, corresponding to W_{up}, W_{act}, and W_{down}, respectively. For DeepSeek, the indices of the removed outliers are listed on top of each graph.

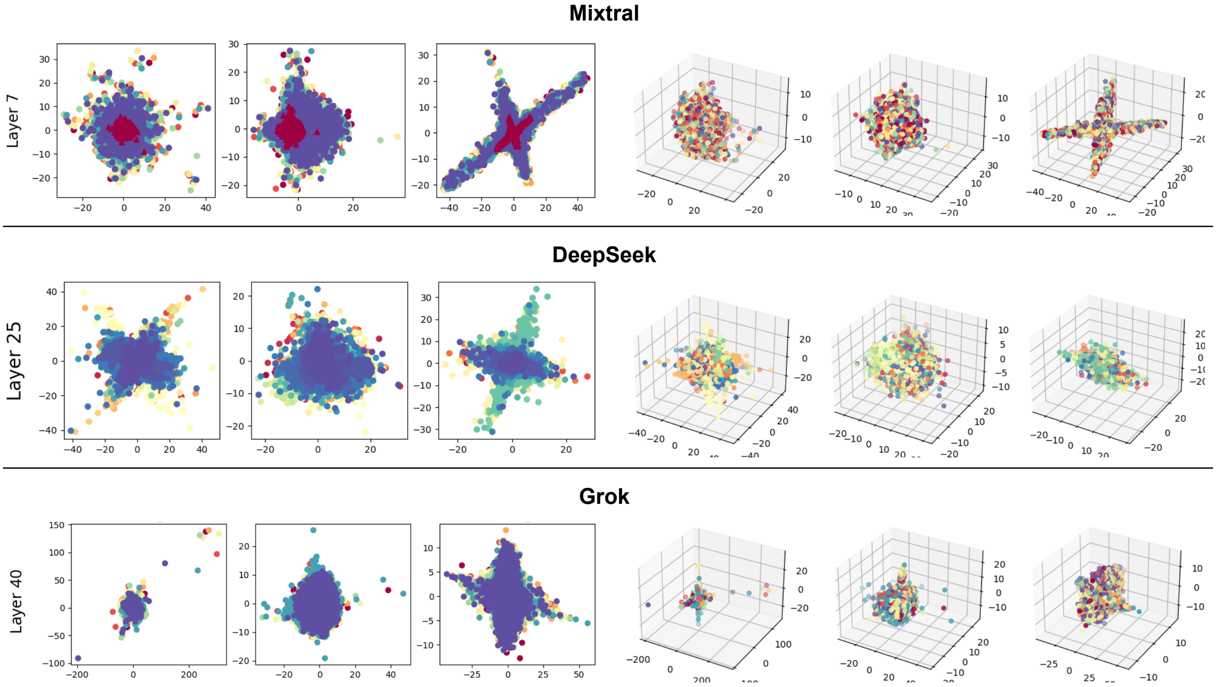

Figure 7: Projection of expert neurons in 2D/3D space. Each layer contains three graphs, corresponding to W_{up}, W_{act}, and W_{down}, respectively.

the similarities of W_{act} significantly increase.

Mixtral and Mistral. The dark crosses sometimes disappear. In the figures for W_{act}, the similarities between the experts and the Mistral FFN are often lower than the similarities among the experts themselves (*i.e.*, $S_{\text{ee}} > S_{\text{ef}}$), which is contrary to previous observations. This can happen if the expert neurons in different positions are alike. For instance, given three vectors $f = (0, 0)$, $e_1 = (1, 0)$, and $e_2 = (0, 1)$, the vector similarity $S_{e_1 e_2}$ is lower than $S_{e_1 f}$ and $S_{e_2 f}$. If averaging the elements, we have $\bar{f} = (0)$, $\bar{e}_1 = (0.5)$, and $\bar{e}_2 = (0.5)$, then $S_{e_1 e_2}$ becomes the highest.

DeepSeek. The growth of the W_{act} similarity values is directly proportional to the layer depth.

Gork. In the heat map of W_{act}, dark crosses frequently appear in various positions.

E Gate Embedding

Since the gate embedding matrix W_g determines the gate decision, there may be a relationship between W_g and the experts. To investigate this, we measure the similarities between the gate embedding vectors, $W_g[n, :]$ for $n \in [1, N]$, and compare them with the neuron-level averaging heat maps of experts presented in § 4.1.2. The qualitative analysis of the combined graphs shown in Fig. 8 is detailed in this section. The table containing the R values for each layer (Tab. 5) is appended at the end.

Mixtral. Focusing on the heat maps of W_g, the similarities typically range from 0.2 to 0.4, with a noticeable increase in the last layer. Moreover, dark crosses are rarely found. Surprisingly, the patterns in the heat maps of W_g and of expert neurons in W_{act} are partially alike in some layers. This implies that the way a gate selects experts might be relevant to how an expert activates its neurons.

DeepSeek. Unlike the almost all-zero heat maps of W_{up} and W_{down}, the similarities of gate neurons sometimes exceed 0.4. In addition, the heat maps of W_g and W_{act} show similar patterns. However, the overall similarities of W_g decrease with depth while the similarities of W_{act} gradually grow. This indicates that as the layer depth increases, the gate "looks" at the input feature in more diverse ways when assigning scores to different experts, even as the neuron activations of the experts become more similar.

Grok. Both dark and bright crosses commonly exist in the heat maps of W_g, whose patterns are

Layer	Mixtral	Mixtral-22	DeepSeek	Grok
0	0.82	-0.01	---	0.89
1	-0.44	0.23	0.75	-0.10
2	0.26	0.62	0.78	-0.28
3	0.54	0.76	0.71	0.66
4	0.48	0.47	0.77	0.52
5	0.70	0.49	0.77	0.37
6	0.84	0.28	0.69	0.28
7	0.74	0.69	0.73	0.17
8	0.42	0.66	0.66	0.51
9	0.66	0.84	0.66	0.84
10	0.53	0.59	0.63	0.28
11	0.32	0.61	0.60	0.30
12	0.14	0.55	0.54	0.46
13	0.51	0.48	0.60	0.14
14	0.66	0.62	0.56	0.00
15	0.40	0.59	0.58	0.54
16	0.39	0.68	0.53	0.32
17	0.53	0.65	0.55	0.30
18	0.35	0.66	0.57	0.10
19	0.17	0.72	0.57	-0.17
20	0.51	0.77	0.58	0.24
21	0.63	0.67	0.62	0.58
22	0.36	0.65	0.62	0.46
23	0.51	0.35	0.62	0.14
24	0.48	0.30	0.68	0.10
25	0.66	0.13	0.62	0.00
26	0.81	0.27	0.58	-0.10
27	0.63	0.22	0.46	-0.26
28	0.73	0.38	---	-0.66
29	0.75	0.52	---	-0.41
30	0.84	0.38	---	-0.83
31	0.57	0.45	---	-0.76
32	---	0.37	---	-0.24
33	---	0.28	---	-0.53
34	---	0.72	---	-0.46
35	---	0.69	---	0.14
36	---	0.34	---	0.17
37	---	0.46	---	-0.46
38	---	0.31	---	-0.17
39	---	0.26	---	-0.26
40	---	0.48	---	-0.70
41	---	0.41	---	0.17
42	---	0.46	---	0.00
43	---	0.33	---	-0.17
44	---	0.44	---	-0.22
45	---	0.50	---	0.14
46	---	0.43	---	-0.47
47	---	0.34	---	-0.44
48	---	0.50	---	-0.17
49	---	0.42	---	-0.14
50	---	0.43	---	0.17
51	---	0.51	---	0.22
52	---	0.67	---	0.10
53	---	0.32	---	0.33
54	---	0.68	---	-0.24
55	---	0.20	---	-0.57
56	---	---	---	-0.24
57	---	---	---	-0.37
58	---	---	---	0.00
59	---	---	---	-0.69
60	---	---	---	-0.17
61	---	---	---	0.35
62	---	---	---	0.30
63	---	---	---	0.10

Table 5: Pearson correlation coefficients (R) of the paired dataset (X, Y_{act}).

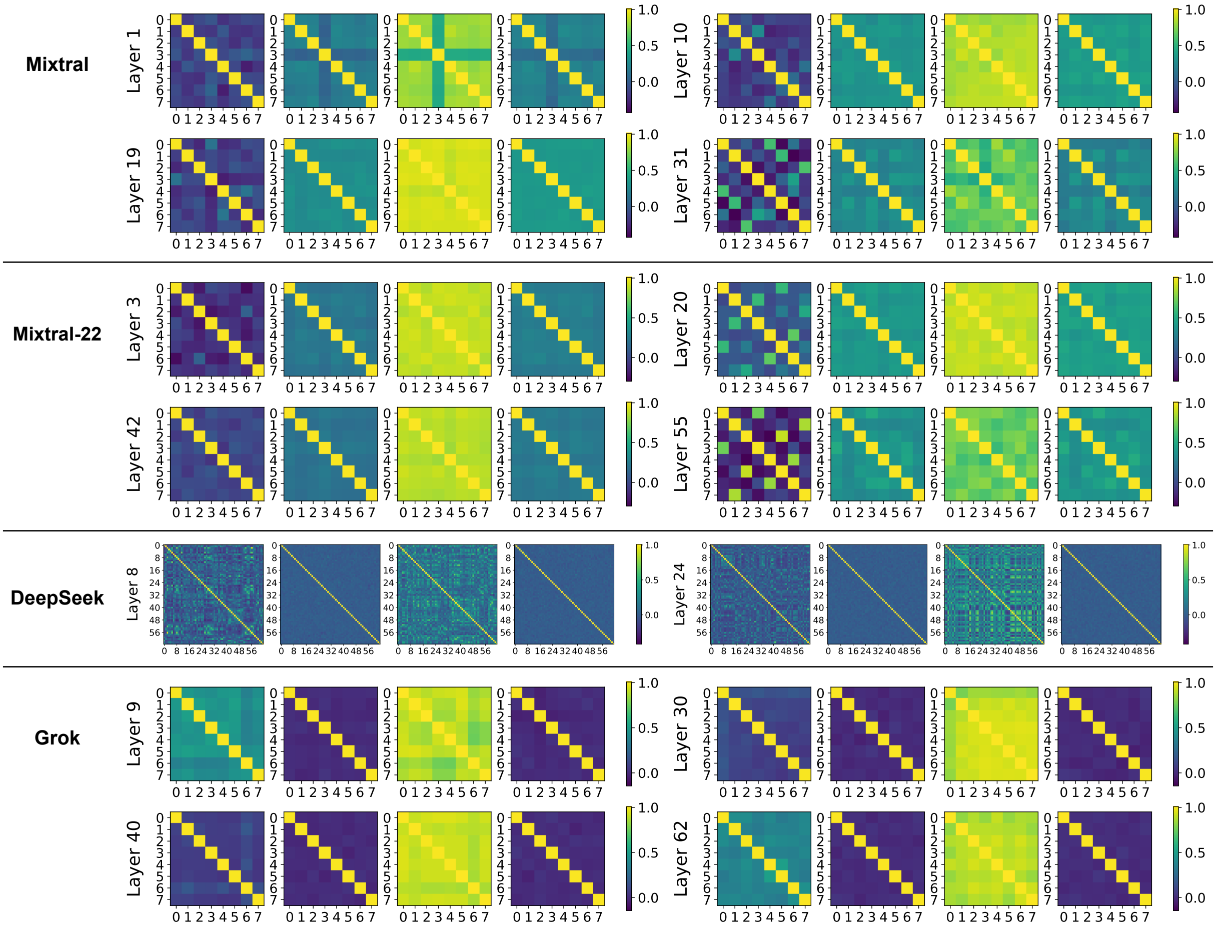

Figure 8: Similarity heat maps of gate embedding (leftmost graph of each layer) along with the neuron-level similarity heat maps using averaging method. The tick numbers refer to the expert indices.

similar to those of W_{act}. Specially, their patterns show opposite color tendency (*i.e.*, deep color positions in one heat map becomes light color in another) starting form the intermediate layers. The similarities of W_g decrease when the layer depth increases, except for the last few layers.

F Additional Datasets

To ensure the universality of our findings, we repeat the experiments that require the long input (§ 5.1, § 5.2) using additional datasets. Specifically, we utilize the entire test set of WikiText-103 (Merity et al., 2016) (266K tokens) and of a math dataset GSM8K (Cobbe et al., 2021) (84K tokens), and 1000 sequences from the code dataset Magicoder-Evol-Instruct-110K (Wei et al., 2024b) (188K tokens). As shown in Fig. 9, 11, 13, the new figures of Mixtral and DeepSeek align with the previous results illustrated in the main context, even when using datasets of specific subjects like math and code (we did not test on the Grok model due to limited computation resources). These supplementary results demonstrate that our findings are general and not limited to the initial input sources.

G Norms of Expert Outputs and Gate Scores

In § 5.2, we notice that in some MoE models, the two experts chosen by the gate usually produce feature vectors with the highest norms. To further investigate this, we repeat the experiment using the long input and additional datasets, and the statistical results are shown in Fig. 10, 11, 13.

Mixtral. It is evident that the expert which outputs the largest norm is most frequently assigned the highest score. Surprisingly, for every i, the i-th highest score is most likely assigned to the expert with the i-th highest output.

DeepSeek. For the experts that generate the first few largest norms (rank 60th to 64th), they are most likely to receive the highest scores. But we do not observe a similar relationship for the rest of the experts. On the contrary, the gate assigns relatively high scores more frequently than low scores to the experts with the smallest norms. For experts ranked 49th to 59th in terms of output norms, they tend to receive either low scores or high scores.

Grok. In contrast to the previous models, the output norms of the Grok experts tend to have an inverse relationship with the scores. More generally, the experts with the first few highest outputs are frequently assigned either low scores or high scores. One possible explanation could be the relatively low activation ratios of GeLU (see Append H), which may result in a weaker dependence on the norm for gate decisions.

H Intermediate States of Experts

While § 5.1 focused on the final outputs of experts, we continue our analysis here by examining their intermediate outputs to examine the inner states of the experts. Given an input x, the intermediate state of an expert refers to the output of $\sigma(W_{\text{act}}x) \in \mathbb{R}^{d_{\text{hid}}}$, where σ denotes an activation function. These intermediate vectors control the activation of neurons, so we simply record them for analysis with the short input used. Mixtral, Mistral, and DeepSeek utilize SiLU as the activation function, while Grok adopts GeLU. Fig. 12 depicts the magnitude of the vectors for Mixtral across three tokens.

Common. Each figure contains some horizontal lines, indicating the presence of an "outlier" expert with either the highest or lowest activation values. Nonetheless, there is no clear relationship between these phenomena and the gate decisions.

Mixtral and Mistral. For a single token, we found that, on average, the absolute activation value of 99.6% elements in each expert exceeds 0.001 after applying the SiLU activation function. This high ratio indicates that the vast majority of neurons in an expert are activated. In Fig. 12, some vertical lines across all experts are commonly found, meaning that the W_{act} matrices of different experts assign similar activation values to neurons with the same indices. In addition, the magnitude of the intermediate states grows along with layer depth, which aligns with the observation in § 5.1.

DeepSeek. On average, 99.7% of the neurons in each expert have an absolute activation value exceeding 0.001 after applying SiLU. Vertical lines rarely exist in the DeepSeek model. Similarly, the elements in the intermediate state vectors get larger as the layer goes deeper.

Grok. With GeLU as the activation function, only 25.3% neurons per Grok expert attain an absolute activation value greater than 0.001. The activation values are generally smaller than those in Mixtral and DeepSeek. Li et al. (2022); Song et al. (2024a) suggest such difference largely stems from the distinct activation functions used. Interestingly, Song et al. (2024b) further utilize the sparsity in experts

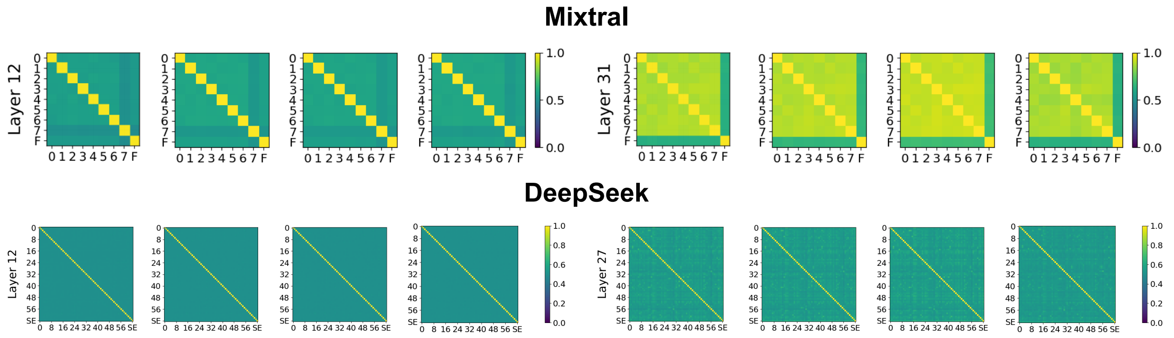

Figure 9: Average similarity heat maps of expert output features using (1) the long sequence, (2) WikiText-103, (3) GSM8K, and (4) Magicoder-Evol-Instruct-110K. The tick numbers refer to the expert indices. “F” and “SE” denote the Mistral FFN and the DeepSeek shared expert, respectively.

non-MoE layer idx	PPL ↓	Bench avg ↑	HellaSwag	MMLU	GSM8K	CEval
none	8.51	39.84	61.75	41.14	13.42	43.04
1	8.50	39.42	61.88	40.55	11.83	43.41
6	8.49	38.55	62.17	39.28	12.96	39.78
12	8.51	38.21	61.46	40.24	10.54	40.61
18	8.59	37.99	61.73	39.50	11.68	39.03
24	8.58	38.94	61.33	40.45	12.05	41.91

Table 6: Model performance on various benchmarks for the dynamic expert numbers experiment. “Bench avg” refers to the average performance over the four evaluated benchmarks.

num_layers	24
vocab_size	151936
hidden_size	1024
head_dim	64
q_head	16
kv_head	4
moe_hidden_dim	640
num_shared_expert	4
num_routed_expert	64
topk	4

Table 7: Model architecture for the dynamic expert numbers experiment.

within SMoE to achieve SOTA performance when activating the same number of parameters.

I Chosen Experts

This experiment aims to examine the routing patterns. We feed an input prompt with about 64 tokens into the MoE models and record the gate scores (after applying softmax) for the selected experts for each token. In addition to the base model of Mixtral (Mixtral-Base), we also include its instruct version (Mixtral-Instruct) in this experiment. The results are depicted in Fig. 14.

Mixtral. In Mixtral-Base, the experts are selected fairly evenly across tokens, and it is common to see sequences of more than four tokens routed to the same expert. But the “special expert” with the dark cross in previous similarity graphs turns out to be an exception. These special experts are chosen less frequently and tend to receive relatively low scores. The routing pattern of Mixtral-Instruct is largely identical to that of Mixtral-Base, which indicates fine-tuning has little impact on gate decisions.

DeepSeek. In some layers, there is an expert selected by most tokens. However, no distinct characteristics for these experts are observed in the previous similarity heat maps. Note that the gate scores for DeepSeek are typically lower than those for Mixtral because DeepSeek applies softmax before the top-k operation, while Mixtral adopts the reverse way.

Grok. The expert selection is rather even and some relatively high scores exist in the deeper (>30th) layers. Same as DeepSeek, softmax is applied before the top-k operation for Grok.

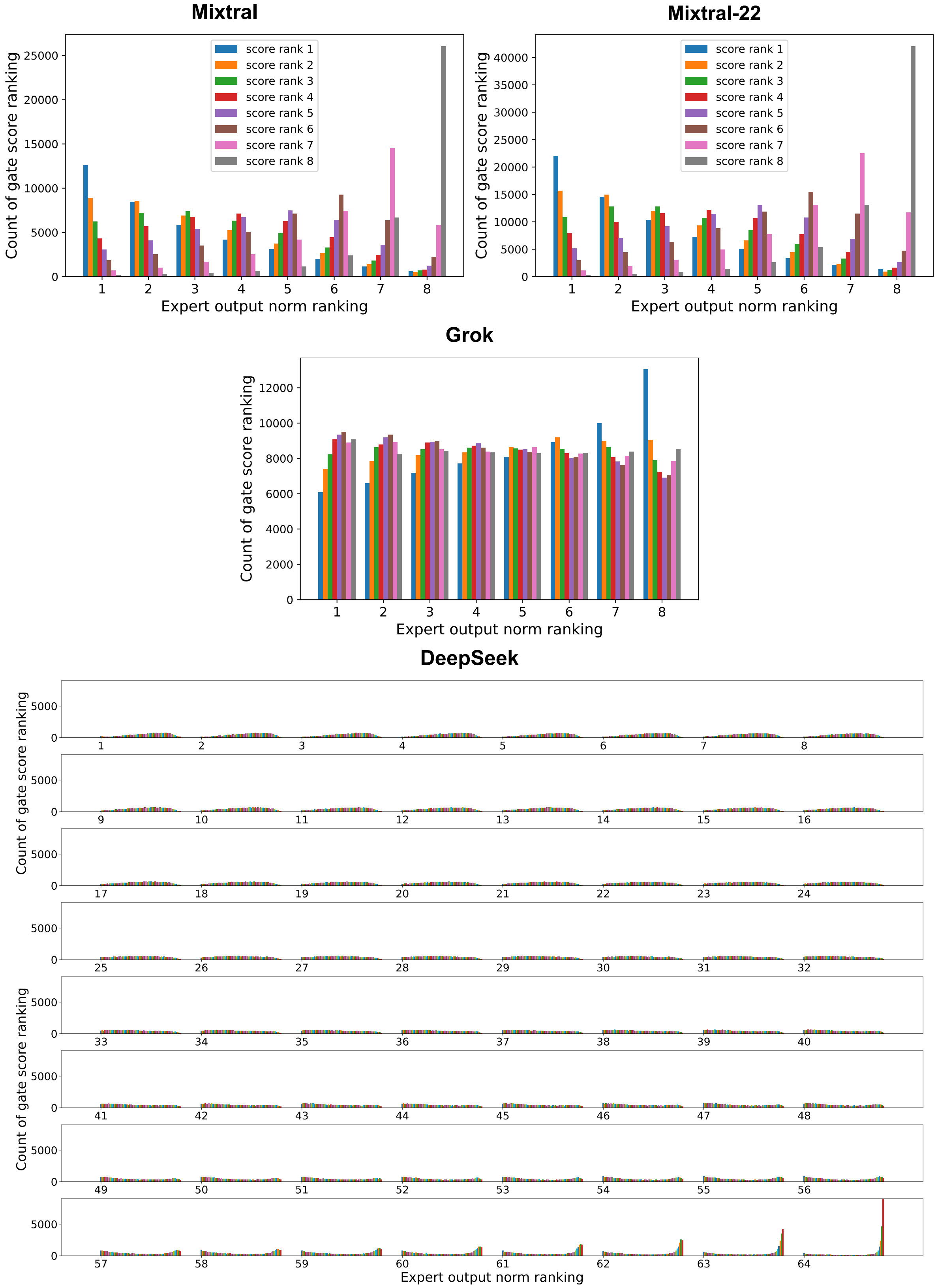

Figure 10: Counts of the gate score ranking for each norm ranking using the long input. The larger the rank number, the larger the norm or score.

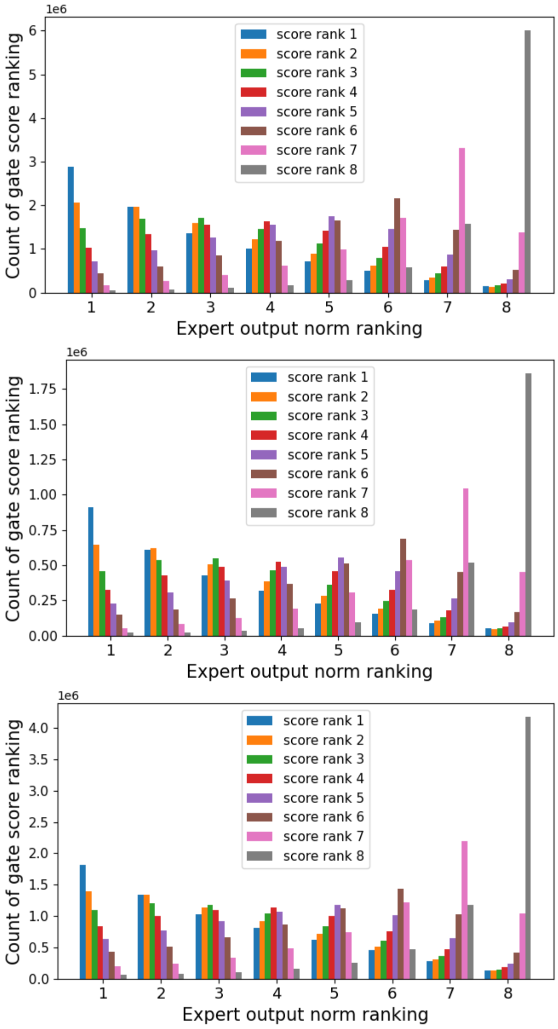

Figure 11: Counts of the gate score ranking for Mixtral expert ourput norm rankings using additional datasets, namely WikiText-103 (**top**), GSM8K (**middle**), and Magicoder-Evol-Instruct-110K (**bottom**). The larger the rank number, the larger the norm or score.

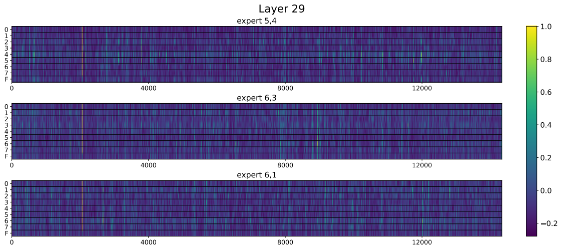

Figure 12: Intermediate state values of Mixtral experts. The top k experts are shown on top of each heat map. Each number in the vertical axis refers to an expert index while the horizontal axis represents the number of neurons.

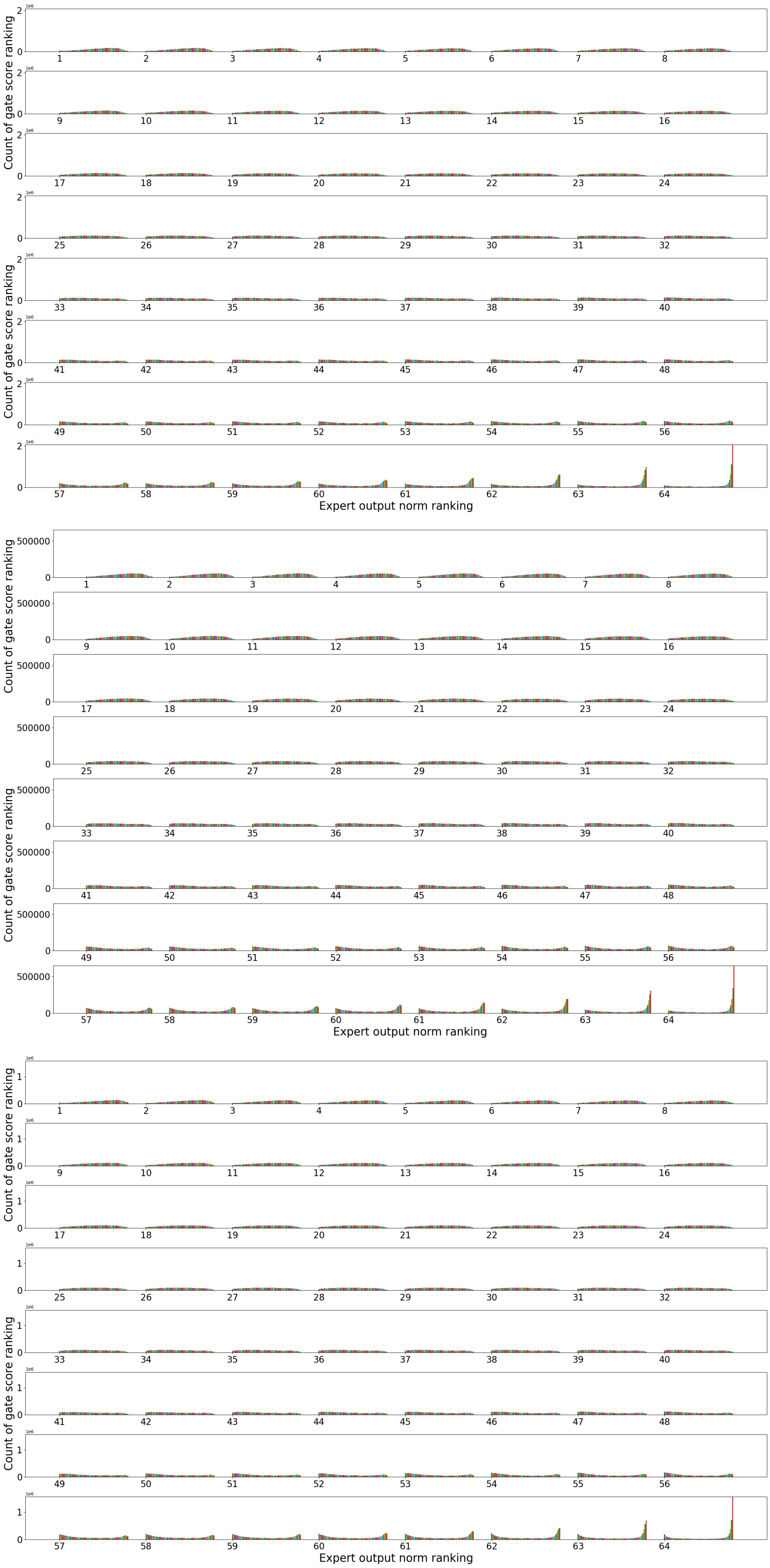

Figure 13: Counts of the gate score ranking for DeepSeek expert ourput norm rankings using additional datasets, namely WikiText-103 (**top**), GSM8K (**middle**), and Magicoder-Evol-Instruct-110K (**bottom**) The larger the rank number, the larger the norm or score.

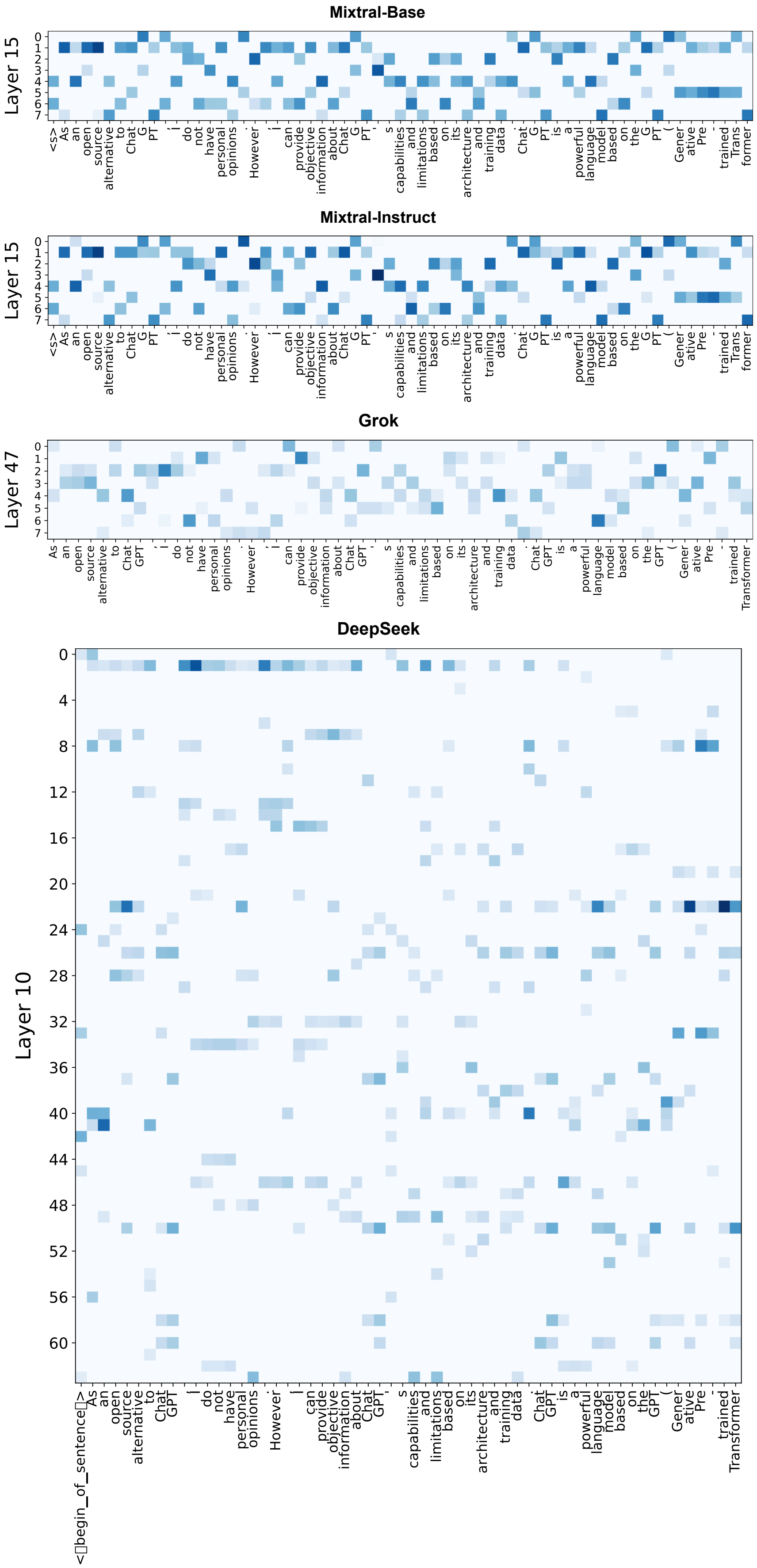

Figure 14: Routing patterns of different models. Deeper colors mean higher gate scores assigned to the corresponding experts. Only scores of the top k experts are illustrated.