Sparse Mixture-of-Experts for Compositional Generalization: Empirical Evidence and Theoretical Foundations of Optimal Sparsity

Jinze Zhao[1], Peihao Wang[1], Junjie Yang[2], Ruisi Cai[1], Gaowen Liu[3],
Jayanth Srinivasa[3], Ramana Rao Kompella[3], Yingbin Liang[2], Zhangyang Wang[1]
[1]University of Texas at Austin
[2]Ohio State University
[3]Cisco Research
{jz24694, peihaowang}@utexas.edu

Sparse Mixture-of-Experts (SMoE) architectures have gained prominence for their ability to scale neural networks, particularly transformers, without a proportional increase in computational cost. Despite their success, their role in compositional generalization, i.e., adapting to novel combinations of known components, remains under-explored. This study challenges the assumption that minimal expert activation suffices for task generalization and investigates the relationship between task complexity and optimal sparsity in SMoE models. Through empirical evaluations on the SRAVEN symbolic reasoning task and the SKILL-MIX benchmark, we demonstrate that (i) the number of activated experts consistently increases with the perceived task difficulty to maintain performance; and (ii) the optimal number of activated experts scales proportionally with task complexity. Our theoretical analysis derives a scaling law for optimal sparsity by balancing approximation and estimation errors, revealing alignment with empirical observations. We formally show that the optimal sparsity lies between minimal activation (1-2 experts) and full activation, with the exact number scaling proportionally to task complexity and further influenced by the size of the training data and the complexity of the model. These findings offer practical insights for designing SMoE models that achieve computational efficiency while enabling robust compositional generalization.

1. Introduction

The Sparse Mixture of Experts (SMoE) model, introduced to modern deep learning first by Shazeer et al. [1], has emerged as a compelling approach for scaling neural networks, particularly transformers, by significantly increasing model size without proportionally increasing computational costs. SMoE achieves this by partitioning the traditional feed-forward network into multiple homogeneous expert networks, dynamically and sparsely activated by a router. This modular architecture not only optimizes computational efficiency but also enhances generalization capabilities, especially in diverse data domains [2]. SMoE has become a standard architecture for many Large Language Models (LLMs) due to its superior performance and scalability [3–9].

Despite these advancements, the application of SMoE models to **compositional generalization** remains underexplored. Compositional generalization tasks require models to solve problems involving novel combinations of familiar components, with difficulty growing exponentially as the design space of possible combinations expands. For instance, in symbolic reasoning, a model may learn to solve individual arithmetic operations like addition and multiplication during training but must generalize to unseen combinations of these operations, such as nested expressions like $(3 + 5) \times 2$, which were not explicitly encountered before. The difficulty scales exponentially as the number of components and their interactions increase. Conventional SMoE configurations typically use minimal expert activation at some fixed, *de-facto* sparsity (e.g., Top 1 or 2 out of 8 experts), a design that may become suboptimal for handling tasks of growing composition complexity. This

Third Conference on Parsimony and Learning (CPAL 2026).

raises a critical question: **Does the *de-facto* activation sparsity remain optimal as task complexity increases in compositional settings?**

To address this, we conduct both theoretical and empirical investigations, demonstrating that the optimal sparsity level for SMoE models scales proportionally with task complexity. Our contributions are summarized as follows:

- We first trained SMoE-based transformers from scratch on the SRAVEN [10] synthetic symbolic reasoning task, varying task difficulty and the number of activated experts. Our findings reveal that activating more experts improves both Out-of-Distribution (OOD) generalization and test accuracy on harder tasks, with the optimal number of activated experts scaling roughly with the task's feature complexity.
- We then evaluated two pretrained SMoE-based LLMs, Mixtral-8×7B [3] and DBRX-132B Instruct [5], on the SKILL-MIX benchmark [11], which challenges models to generate coherent text that integrates k linguistic skills. Results show that activating more experts-per-token notably improves performance on harder tasks without additional training.
- Built on consistent empirical observations, we derived a theoretical scaling law for optimal sparsity in SMoE models under compositional input data settings, under simplified assumptions. Our analysis reveals that the optimal sparsity lies between minimal activation (e.g., 1-2 experts) and full activation of all experts, with the precise number depending on task difficulty, the size of the training data, and the complexity of the model.

Our work challenges the prevailing assumption that minimal expert activation is sufficient for complex tasks. By identifying the relationship between task difficulty and optimal sparsity, we provide actionable insights into designing SMoE models that balance computational efficiency with robust compositional generalization.

2. Related Works

Compositional Generalization Compositional generalization refers to a system's ability to understand and generate novel combinations of a finite set of familiar elements [12, 13]. This capability is essential for enabling efficient learning and achieving robust generalization across diverse domains. In the computer vision field, studies have focused on generating images from new concept combinations, often using disentangled representation learning [14]. Researchers have evaluated VAE-based generative models in compositional tasks [15, 16], exploring the relationship between disentanglement and generalization performance in image reconstruction and generation. Recent studies have made significant strides by conducting a controlled investigation of compositional generalization in conditional diffusion models [17–19], revealing insights into the emergence of compositional abilities and the factors influencing out-of-distribution generation.

Recent works have observed emergent compositional capabilities in LLMs [11, 20, 21]. Several evaluation methods have been proposed to quantify compositional generalization of large, *monolithic* pre-trained LLMs, including imposing generation constraints [11], multi-hop question answering [22], and elementary math operations [23]. Theoretical advancements have also shed light on the conditions required for achieving compositional generalization in neural networks [10, 24, 25]. More recently, Huang et al. [26] finds out that harder tasks need more experts and then proposed a heterogeneous routing strategy, and Abnar et al. [27] investigates the sparsity scaling-law of SMoE, though both from non-compositional settings and without theoretical analysis. Despite substantial progress, there is a significant gap in understanding *compositional generalization within modular architectures*, particularly SMoEs. Our work pioneers this investigation by exploring whether *de-facto* sparse activation remains optimal as compositional task difficulty increases from both theoretical and experimental perspectives.

Theoretical Understanding of Mixture-of-Experts Recent work [28] formally studied how the SMoE layer reduces training error more effectively than a single expert and why such a mixture model does not collapse into a single model. Importantly, when training an SMoE layer on data generated from a 'mixture of class' distribution using gradient descent, the authors proved that each expert in the SMoE model specializes in a specific portion of the data (at least one cluster), while the router learns the cluster-center features and routes inputs to the appropriate experts. Subsequently, a series of studies [29–33] sought to establish the convergence rates of density estimation and parameter estimation in MoE models by defining Voronoi-based losses that describe the interaction between the gating function and experts, explaining why Top-1 gating enables faster convergence rates for parameter estimation compared to other gating mechanisms. More recently, Jelassi et al. [34] theoretically justified that while the memorization performance consistently improves with an increasing number of experts in SMoE, reasoning capabilities tend to saturate. Meanwhile, Akretche et al. [35] provided tighter risk bounds by incorporating local differential privacy into the gating mechanism to enhance the generalization ability of MoE. In contrast to those prior arts, we aim to formally study whether the conventional sparse activation of SMoE remains an optimal strategy for compositional tasks of increasing difficulty, using both theoretical and empirical approaches.

3. Empirical Study

We postpone the introduction of the used compositional tasks to Section A.1 due to page limit. We conduct two sets of experiments to investigate whether sparse expert activation remains optimal for handling compositional tasks of varying difficulty, in both training-from-scratch and testing pre-trained model settings. In training-from-scratch experiments, we trained and tested SMoE under varying experts sparsity levels and compositional task difficulty levels. We test pretrained SMoE-based LLMs with varying experts sparsity levels and compositional task difficulty levels since they are pre-trained under a fixed sparsity level, *e.g.*, Top-2 routing. Both set of experiments indicate that the *de-facto* activate sparsity is non-optimal as the task complexity increases.

3.1. Training and Evaluting SMoE-based Transformers on SRAVEN

We trained standard decoder-only SMoE-based transformers on SRAVEN synthetic tasks [10]. Each transformer block consists of multi-head attention with relative positional encoding [36] and feed-forward layer (FFN). The feedforward layer in each block is an SMoE structure with 8 parallel homogeneous experts, where each expert is a 2-layer multi-layer perceptron (MLP). The router is a simple 1-layer dense layer with top-K softmax gating mechanism. For the SRAVEN hyperparameters, we fixed the grid size of the problem to be 3×3, which means that we fix the number of in-context examples for the model. We have $R = 8$ possible rules to sample from. Following Schug et al. [10], we split all possible rule combinations at difficulty level M into training and testing set, also hold out 25% of all possible combinations as OOD evaluation set. We can adjust the difficulty of the task by sampling $M \in \{1, 2, \cdots, R\}$ different rules to compose the task.

3.1.1. Sparse activation levels are not consistently optimal across all difficulty settings

We trained SMoE transformers with various Top-K routing mechanisms for the same number of iterations across different difficulty levels of the SRAVEN task, keeping all other model and training hyperparameters constant. Surprisingly, Top-1 routing consistently achieved the worst Out-of-Distribution (OOD) accuracy across all difficulty levels. While Top-2 routing performed comparably to other more expensive routing mechanisms on easier tasks, its performance also declined significantly as the compositional task complexity increased, as shown in Figure 1. Similar trends were observed in Test Accuracy evaluations, detailed in Figure 2 and Section A.2.2. These findings demonstrate that commonly-used “de-facto" sparse activation mechanisms are suboptimal for learning compositional tasks, challenging previous conclusions from studies such as Shazeer et al. [1], Jiang et al. [3].

3.1.2. The optimal number of activated experts roughly scales with task difficulty

We also observe that as we increase the compositional task difficulty, *i.e.*, the number of sampled rules M, more activated experts are required to obtain the optimal performance correspondingly. Starting from the setting where $M = 4$, the optimal number of activated experts K is roughly scaled to M, and the performance gap between each activation mechanism also widens, as shown in Figure 1 and Figure 2. Therefore, we hypothesize that each expert can specialize in specific rules when the model is trained on compositional tasks, a phenomenon not typically observed in traditional NLP training tasks [37].

We also conducted ablation studies by switching from Softmax attention to HYLA attention [10], a novel mechanism that encourages compositional generalization ability, and observed similar results

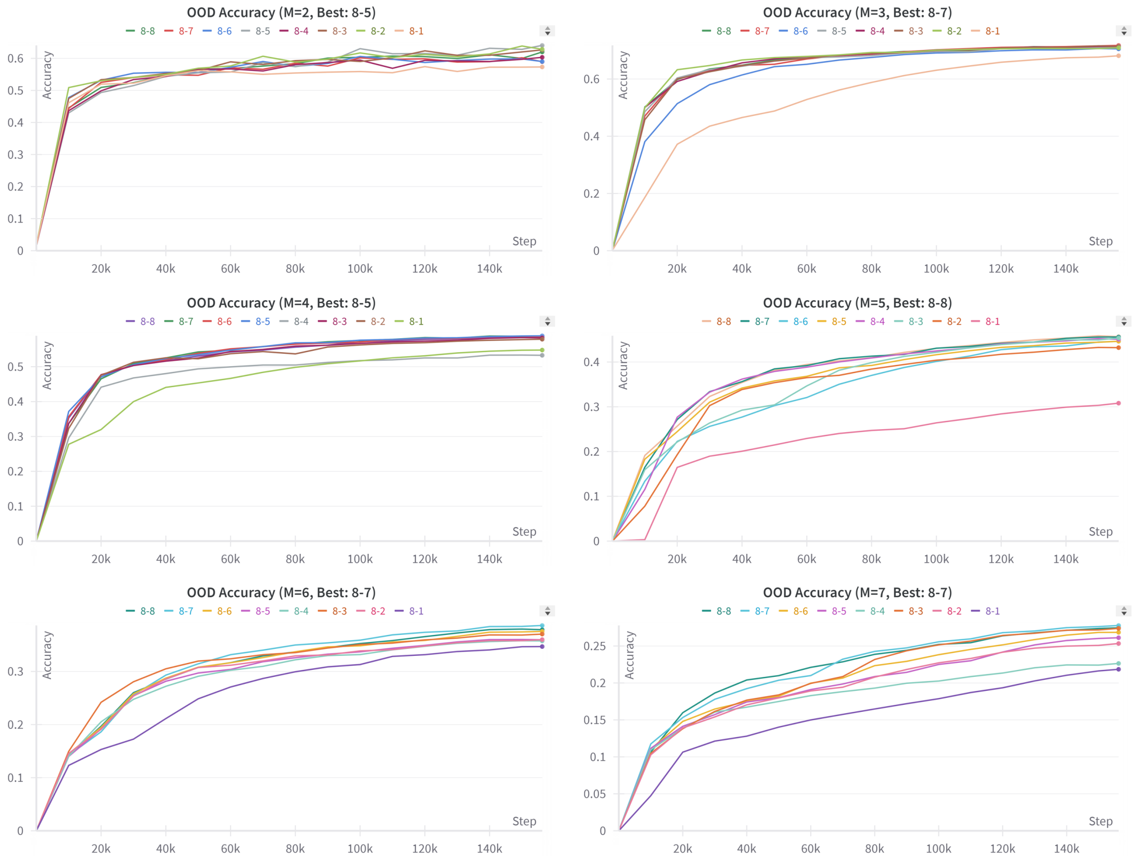

Figure 1: OOD accuracy of training SMoE Transformer, where the difficulty level of the task (i.e., the number of sampled rules M) increases. ”8-k” means activating k out of 8 experts on every FFN layer. The best-performing activation mechanism is labeled on the caption of each figure.

as shown in Section A.2.1. These observations collectively suggest that the choice of attention mechanism and the number of activated experts must be dynamically adjusted based on task difficulty. Specifically, HYLA attention coupled with increased expert activation is promising for robust compositional generalization in challenging tasks, while simpler tasks may not benefit significantly from these adjustments.

3.2. Evaluating Pre-trained SMoE-based LLMs on Skill-Mix

We evaluate two instruction-tuned SMoE-based LLMs, *i.e.*, Mixtral-8×7B Instruct-v0.1 [3] and DBRX-132B Instruct [5], on the Skill-Mix Yu et al. [11] benchmark. These evaluations utilize pre-trained models, meaning the training-time number of experts-per-token is already fixed and remains unchanged. Therefore, our experiments test the impact of varying the inference-time experts-per-token, essentially exploring the Out-of-Distribution (OOD) generalization of the number of active experts during the test phase. To ensure consistency in evaluation, we adopt GPT-4 [38] as the grading model and use the released 10% of the skill and topic lists from Section A of Yu et al. [11]. The same optimized generation prompts are used for querying Mixtral and DBRX, and identical grading prompts and evaluation metrics are used to query GPT-4 for grading, as outlined in Yu et al. [11].

3.2.1. More Experts-per-token Improves Compositional Generalization on Harder Tasks

We evaluate Mixtral-8×7B Instruct-v0.1 on Skill-Mix by varying both the task difficulty k (i.e., the number of skills the model must combine in its generated response) and the number of experts-per-token (ept) during inference. Similar patterns are observed in the DBRX evaluation, as summarized in Table 1 and Table 2. Key findings include:

- **Scaling Experts with Task Difficulty:** As the difficulty of the Skill-Mix task increases, activating more experts per token during inference is essential for performance. For example, the generated outputs from the default Top-2 routing setting score zero on all grading metrics when $k = 4$, indicating that two experts are insufficient for such complex compositional tasks. In contrast, activating 4 or 5 experts achieves the best performance for $k = 4$.

- **Optimal Experts Scale with Task Complexity:** The optimal number of experts per token scales with the task difficulty k. For simpler tasks (e.g., $k = 1$ or $k = 2$), a smaller number of experts (e.g., Top-2) suffices, while harder tasks (e.g., $k = 4$) require more experts (e.g., 4 or 5 experts per token) to maintain compositional generalization.

- **Over-activation Can Harm Simpler Tasks:** Unlike the SRAVEN training experiments, which showed that increasing the number of activated experts does not degrade performance on simpler compositional tasks, the Skill-Mix evaluation of pre-trained LLMs highlights a downside of over-activation during inference. For instance, while the default Top-2 routing achieves perfect scores for $k = 1$ and $k = 2$ tasks, activating 7 or 8 experts during inference significantly lowers the model's performance on these simpler tasks, likely due to unnecessary complexity introduced by activating more experts than required.

These findings demonstrate that the optimal number of experts-per-token during inference is task-dependent, with harder tasks requiring more active experts for better compositional generalization. However, overactivating experts can negatively impact simpler tasks, highlighting the importance of dynamically adapting the number of experts based on task complexity. This insight emphasizes the need to treat training-time and inference-time sparsity settings separately to achieve robust performance across a range of task difficulties.

Skill-Mix results for Mixtral-8×7B evaluated by GPT4	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
EPT=1	0.00 ± 0.000 0.00 ± 0.000 0.00 ± 0.000	0.00 ± 0.000 0.00 ± 0.000 0.00 ± 0.000	0.00 ± 0.000 0.00 ± 0.000 0.20 ± 0.082	0.00 ± 0.000 0.00 ± 0.000 0.35 ± 0.100	0.00 ± 0.000 0.00 ± 0.000 0.00 ± 0.000
EPT=2 (default)	**1.00 ± 0.000** **1.00 ± 0.000** **1.00 ± 0.000**	**1.00 ± 0.000** **1.00 ± 0.000** **1.00 ± 0.000**	0.00 ± 0.000 **1.00 ± 0.000** 0.00 ± 0.000	0.00 ± 0.000 0.00 ± 0.000 0.00 ± 0.000	0.00 ± 0.000 0.00 ± 0.000 0.00 ± 0.000
EPT=3	0.00 ± 0.000 0.40 ± 0.245 0.00 ± 0.000	0.00 ± 0.000 0.20 ± 0.200 0.10 ± 0.100	0.00 ± 0.000 0.00 ± 0.000 0.47 ± 0.082	0.00 ± 0.000 0.00 ± 0.000 0.60 ± 0.061	0.00 ± 0.000 0.00 ± 0.000 0.00 ± 0.000
EPT=4	0.20 ± 0.200 0.80 ± 0.200 0.20 ± 0.200	0.20 ± 0.200 0.40 ± 0.245 0.40 ± 0.187	**0.20 ± 0.200** **0.20 ± 0.200** **0.67 ± 0.105**	**0.20 ± 0.200** **0.20 ± 0.200** **0.75 ± 0.079**	0.00 ± 0.000 **0.20 ± 0.200** 0.00 ± 0.000
EPT=5	0.20 ± 0.200 0.40 ± 0.245 0.20 ± 0.200	0.20 ± 0.200 0.20 ± 0.200 0.30 ± 0.200	0.20 ± 0.200 0.20 ± 0.200 0.53 ± 0.133	**0.20 ± 0.200** **0.20 ± 0.200** **0.65 ± 0.100**	0.00 ± 0.000 **0.20 ± 0.200** 0.00 ± 0.000
EPT=6	0.00 ± 0.000 0.20 ± 0.200 0.00 ± 0.000	0.00 ± 0.000 0.00 ± 0.000 0.20 ± 0.122	0.00 ± 0.000 0.00 ± 0.000 0.47 ± 0.082	0.00 ± 0.000 0.00 ± 0.000 0.60 ± 0.061	0.00 ± 0.000 0.00 ± 0.000 0.00 ± 0.000
EPT=7	0.00 ± 0.000 0.00 ± 0.000 0.00 ± 0.000	0.00 ± 0.000 0.00 ± 0.000 0.10 ± 0.100	0.00 ± 0.000 0.00 ± 0.000 0.40 ± 0.067	0.00 ± 0.000 0.00 ± 0.000 0.55 ± 0.050	0.00 ± 0.000 0.00 ± 0.000 0.00 ± 0.000
EPT=8	0.00 ± 0.000 0.40 ± 0.245 0.00 ± 0.000	0.00 ± 0.000 0.20 ± 0.200 0.10 ± 0.100	0.00 ± 0.000 0.00 ± 0.000 0.47 ± 0.082	0.00 ± 0.000 0.00 ± 0.000 0.60 ± 0.061	0.00 ± 0.000 0.00 ± 0.000 0.00 ± 0.000

Table 1: Skill-Mix Evaluation Results on Mixtral-8×7B Instruct-v0.1 [39]. The grading metrics are Ratio of Full Marks/Ratio of All Skills/Skill Fraction as defined in Section A.3.1. 'EPT' stands for 'Number of experts per token.'

4. Theoretical Analysis

In this section, we provide a novel theoretical analysis to justify our empirical findings. We derive the generalization and approximation errors for SMoE trained on compositional tasks. Based on this result, we show a theoretical trade-off between routing sparsity and task complexity, aligned with our empirical observations.

4.1. Preliminaries

To begin with, we formally introduce the necessary mathematical setups for our successive analysis.

Compositional Learning. Consider a data distribution: $\mathcal{D} \sim \mathbb{P}(\boldsymbol{x}, y)$, where $\boldsymbol{x} \in \mathbb{R}^d$ is an input feature vector and $y \in \mathcal{Y} \subseteq \mathbb{R}$ is the corresponding label. In compositional learning, we assume distribution $\mathcal{D}$ consists of N tasks, and there are N corresponding skills to solve each task, accordingly. For each input $\boldsymbol{x}$, we assume it can be represented by a subset of tasks and solved by corresponding skills, and its label y is synthesized by combining the results of the corresponding skills. Mathematically, we denote skills as a set of functions $\{t_1, \cdots, t_N\}$. The data distribution is generated by $\boldsymbol{x} \sim \mathbb{P}(\boldsymbol{x})$, $y = G_{\mathcal{I}(\boldsymbol{x})}(t_1(\boldsymbol{x}), \cdots, t_N(\boldsymbol{x}))$, where $\mathcal{I} \in 2^{[N]}$ indicates the task assignment of the input $\boldsymbol{x}$, and $G_{\mathcal{I}}$ is the composition function according to $\mathcal{I}$. We assume our training dataset $\mathcal{S}$ consists of m i.i.d. samples drawn from the compositional data distribution: $\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_m, y_m)\} \overset{i.i.d.}{\sim} \mathcal{D}^m$. Given a parametric learner $f : \mathbb{R}^d \to \mathcal{Y}$, we optimize f to minimize the empirical loss over the training set $\mathcal{S}$: $L_{\mathcal{S}}(f) = \frac{1}{m}\sum_{i=1}^{m} \ell(f(\boldsymbol{x}_i), y_i)$, where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is the element-wise loss function. At the test stage, we consider the error within the whole data domain: $L_{\mathcal{D}}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \ell(f(\boldsymbol{x}), y)$.

Sparse Mixture-of-Experts. We consider Sparse Mixture-of-Expert (SMoE) as the learner to the compositional task defined above. SMoE can be defined as a data-dependent ensemble of many expert networks. Suppose the total number of experts is T and the number of dynamically activated experts is k. Then SMoE can be written as a function $f : \mathbb{R}^d \to \mathcal{Y}$ defined as below:

$$f(\boldsymbol{x}) = \sum_{j=1}^{T} a(\boldsymbol{x})_j h_j(\boldsymbol{x}) \quad \text{subject to} \sum_{j=1}^{T} \mathbb{1}\{a(\boldsymbol{x})_j \neq 0\} = k, \quad \forall \boldsymbol{x} \in \mathbb{R}^d, \tag{1}$$

where $a(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^T$ is named as the routing function satisfying $\|a(\boldsymbol{x})\|_0 = k$, and $h_j(\boldsymbol{x}) : \mathbb{R}^d \to \mathcal{Y}$ is an expert network for every $j = 1, \cdots, T$ [1]. Intuitively, $a(\boldsymbol{x})$ selects k experts to be activated to inference labels for $\boldsymbol{x}$. In this paper, we consider the normalized gating function, which first chooses the k logits from the output, sets the remainders to zeros, and applies softmax to normalize the chosen entries:

$$a(\boldsymbol{x})_j = \begin{cases} \frac{\exp(g(\boldsymbol{x})_j)}{\sum_{t\in\mathcal{J}(\boldsymbol{x})} \exp(g(\boldsymbol{x})_t)} & \text{if } j \in \mathcal{J}(\boldsymbol{x}) \\ 0 & \text{if } j \notin \mathcal{J}(\boldsymbol{x}) \end{cases}, \tag{2}$$

where $g : \mathbb{R}^d \to \mathbb{R}^T$ computes the weight for each expert, and $\mathcal{J}(\boldsymbol{x})$ finds a sparse mask with at most k non-zero entries according to $\boldsymbol{x}$, i.e., $|\mathcal{J}(\boldsymbol{x})| = k, \forall \boldsymbol{x} \in \mathbb{R}^d$. Most typically, $\mathcal{J}(\boldsymbol{x})$ selects the indices corresponding to the top-k largest logits from $g(\boldsymbol{x})$ [1].

Now we can formally state the hypothesis space of an SMoE model, which is a composition of both the hypothesis spaces of gating and expert networks.

Definition 1. *Suppose all expert networks $h_1, \cdots, h_T \in \mathcal{H}$ is selected from the same hypothesis space $\mathcal{H}$, and k-sparse routing function $a \in \mathcal{A}$ is chosen from the hypothesis space $\mathcal{A}$. Define the hypothesis space of the SMoE model with T experts and k-sparse routing function following Eq. 1 and Eq. 2 as below:*

$$\mathcal{F}(T, k) = \left\{ f(\boldsymbol{x}) = \sum_{j=1}^{T} a(\boldsymbol{x})_j h_j(\boldsymbol{x}) : h_1, \cdots, h_T \in \mathcal{H}, a \in \mathcal{A} \right\} \tag{3}$$

[1] For simplicity, We assume the (convex) linear combination of $\mathcal{Y}$ is still in $\mathcal{Y}$.

Complexity Metrics. We will also employ two complexity metrics classical in learning theory to characterize the generalization error of SMoE. First, we consider Rademacher complexity, which directly measures the capacity of a model class in terms of its ability to fit random labels, and it can depend on the specific data distribution. Given a class of functions $\mathcal{H}$, data distribution $\mathcal{D}$ and samples $S = \{z_1, \ldots, z_m\}$ drawn i.i.d. from $\mathcal{D}$, we can define Rademacher complexity as below:

Definition 2 (Rademacher complexity)**.** *The empirical Rademacher complexity of $\mathcal{H}$ is defined to be*

$$\mathcal{R}_m(\mathcal{H}, \mathcal{S}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{H}} \left(\frac{1}{m} \sum_{i=1}^{m} \sigma_i f\left(z_i\right) \right) \right] \tag{4}$$

where $\sigma_1, \ldots, \sigma_m$ are independent random variables uniformly chosen from $\{-1, 1\}$. We will refer to such random variables as Rademacher variables. The Rademacher Complexity of $\mathcal{H}$ is defined as $\mathcal{R}_m(\mathcal{H}) = \mathbb{E}_{\mathcal{S} \sim \mathcal{D}^m} \left[\mathcal{R}_m(\mathcal{H}, \mathcal{S})\right]$.

We also consider a combinatorial measure of model capacity – *Natarajan dimension*, which often provides tighter bounds for finite decision regions [40]. Natarajan dimension is a generalization of the VC dimension to classes of multiclass predictors [41]. We formally state it below, which requires us to first generalize the definition of shattering.

Definition 3 (Natarajan Dimension)**.** *We say that a set $\mathcal{C} \subset \mathcal{X}$ is shattered [41] by hypothesis space $\mathcal{H}$ if there exist two functions $f_0, f_1 : \mathcal{C} \to [k]$ such that i) for every $\boldsymbol{x} \in \mathcal{C}$, $f_0(\boldsymbol{x}) \neq f_1(\boldsymbol{x})$; ii) for every $\mathcal{B} \subset \mathcal{C}$, there exists a function $h \in \mathcal{H}$ such that $\forall \boldsymbol{x} \in \mathcal{B}, h(\boldsymbol{x}) = f_0(\boldsymbol{x})$ and $\forall \boldsymbol{x} \in \mathcal{C} \backslash \mathcal{B}, h(\boldsymbol{x}) = f_1(\boldsymbol{x})$. The Natarajan dimension of $\mathcal{H}$, denoted by $\mathrm{NDim}(\mathcal{H})$, is the maximal size of a shattered set $\mathcal{C} \subset \mathcal{X}$.*

When $k = 2$, this definition degenerates to the VC dimension. Interested readers are referred to Shalev-Shwartz and Ben-David [41] for more details.

4.2. Generalization Error Analysis

In this section, we analyze the generalization error defined as the discrepancy between the empirical and population losses: $|L_{\mathcal{S}} - L_{\mathcal{D}}|$.

First of all, we quantify the complexity of a family of routing functions. Our approach is to disentangle the gating output by normalized weights multiplied with a mask: $a(\boldsymbol{x}) = m(\boldsymbol{x}) \odot \boldsymbol{\nu}(\boldsymbol{x})$, where $m(\boldsymbol{x})_j = 1$ if $a(\boldsymbol{x})_j \neq 0$, otherwise $m(\boldsymbol{x})_j = 0$. We note that $m(\boldsymbol{x}) : \mathbb{R}^d \to \{0, 1\}^T$ is a multi-class classifier. Henceforth, we can characterize the complexity of $m(\boldsymbol{x})$ via the Natarajan dimension. Define a family of masking functions $\mathcal{M}$ induced by the class of gating functions $\mathcal{A}$. Then the complexity of $\mathcal{M}$ is specified by the following assumptions.

Assumption 1. *The Natarajan dimension of $\mathcal{M}$ is scaled with the number of tasks N as $\mathrm{NDim}(\mathcal{M}) = O(Nd_N)$ where d_N is the base case when $N = 1$.*

$\mathrm{NDim}(\mathcal{M})$ represents the maximal cardinality of a set that shatters all possible outcomes of $m(\boldsymbol{x})$, which can be used for counting the number of sparse patterns produced by $a(\boldsymbol{x})$ In Assumption 1, the capacity of the sparse router is growing with the number of tasks because more tasks often cause higher complexity of compositional data distribution, which yields more diverse expert combinations to solve the compositional tasks.

Next, we make basic assumptions on the loss function:

Assumption 2. *The loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ is C-Lipschitz.*

Such an assumption is standard in generalization error analysis [41]. Essentially, we hypothesize that the loss function is bounded and Lipschitz continuous, which is satisfied by many common choices (e.g. cross-entropy or MSE) when the inputs or outputs are restricted to a compact space.

Now we can state the main result bounding the generalization error of SMoE under the compositional learning setting:

Theorem 1. *Consider the hypothesis space $\mathcal{F}(T,k)$ stated in Definition 1, under Assumptions 1 and 2, with probability at least $1-\delta$ over the selection of training samples, the generalization error is upper bounded by:*

$$|L_{\mathcal{S}} - L_{\mathcal{D}}| = O\left(4C\mathcal{R}_m(\mathcal{H}) + 2\sqrt{\frac{2kNd_N \log T + Nd_N \log(2m) + \log(2/\delta)}{2m}}\right), \tag{5}$$

where $\mathcal{R}_m(\mathcal{H})$ is the Rademacher complexity of the expert hypothesis space $\mathcal{H}$ (cf. Definition 2).

The complete proof is deferred to Appendix B. Theorem 1 reveals that SMoE can be helpful for compositional learning. This can be seen in the comparison with the naive ensemble, where T weak models are stacked to form the stronger model. Intuitively, the entire SMoE model contains T copies of expert learners, thus, its capacity is as large as the naive ensemble. However, the generalization error of the naive ensemble follows the classical results and gives a rate growing linearly with the number of total experts $O(T\mathcal{R}(\mathcal{H}) + \sqrt{\log(2/\delta)/2m})$, while SMoE only exhibits a logarithmic dependency on T and linear dependency on k. This suggests that dynamic routing not only reduces the inference cost but also shrinks the average model complexity. From the perspective of each data point, only a part of the model is used.

Nevertheless, dynamic routing incurs additional costs. Specifically, it introduces an extra term Nd_N in the error bound due to the flexibility provided by the learnable router. Furthermore, the composition of multiple tasks exacerbates the bound, as the increased task complexity demands a greater variety of routing patterns, making SMoE harder to generalize.

4.3. Optimal Sparsity Analysis

In this section, we aim to analyze the optimal sparsity k the model should choose to obtain the least error. We first decompose the generalization error obtain from Theorem 1 into a different form as the summation of **Approximation Error** and **Estimation Error** under moderate assumptions, where both error terms are derived independently based on sparsity and input task complexity, and then analyze the optimal choice of k under different hyperparameter settings.

4.3.1. Approximation Error Construction.

In this section, we outline the assumptions required to characterize the Approximation Error first, and then construct the Approximation Error.

Assumption 3. *Each input $\boldsymbol{x}$ is a composition of two tasks sampled from a total number of N tasks, i.e., $\boldsymbol{x} = T_i \circ T_j$ where $T_i, T_j \in \{T_1, T_2, \dots, T_N\}$.*

Assumption 4. *Each expert h_i is uniformly assigned with non-overlapping compositional tasks.*

Assumption 5. *Each single task T_i is equally weighted and does not account for task-specific variability such as importance or difficulty of each task.*

We adopt these simplified settings to help with the theoretical analysis. A more complex and realistic *Power-Law Distribution* [42] setting for compositional tasks is described in Section A.5. Under the current simplified setting, we define the Approximation Error of SMoE as following:

Definition 4 (Approximation Error)**.** *Under Assumptions 3 4 5, the Approximation Error of SMoE is modeled as:*

$$E_{\textit{approximation}}(k, N) = O\left(\frac{N^2}{k}\right), \tag{6}$$

where N^2 is the total number of pairwise task compositions, and k is the total number of selected experts.

Remark 2. *The Approximation Error $O\left(\frac{N^2}{k}\right)$ is independent of the number of input data m and it treats each task uniformly and distributes each experts' capacity evenly across all N^2 possible combinations. This error term decreases monotonically as k increases, and when k is small it dominates due to insufficient expert capacity for task combinations.*

4.3.2. Estimation Error Construction.

In this section, we construct the Estimation Error based on Theorem 1 to capture the variations in the generalization error given different sparsity levels, number of input data, and the complexity of the router.

Definition 5 (Estimation Error)**.** *Under Assumptions 3 4 5, the Estimation Error of SMoE is modeled as:*

$$E_{\textit{estimation}}(k, N) = O\left(\sqrt{\frac{Nkd_N}{m}\log\left(\frac{T}{k}\right)}\right),$$

where N, k, d_N, m, and T follows their definitions in Section 4.1.

Remark 3. *Regarding the Estimation Error, we observe that it is a concave function with respect to k, where the error increases and reaches the maximum at $k = \frac{T}{e}$, and then decreases as $k \to T$. The presence of N penalizes large k more strongly, making the estimation error scale with both the number of tasks N and the number of active experts k.*

4.3.3. Bias-Complexity Trade-off

In this section, we provide insights on how does the total error $E_{\text{total}}(k, N) = O\left(\frac{N^2}{k}\right) + O\left(\sqrt{\frac{Nkd_N}{m}\log(T/k)}\right)$ scales with different parameter settings, helping us choose the optimal k^* to minimize the overall error.

k **Dependence (Number of Selected Experts Per Task):** The behavior of the error terms varies significantly with k. For small k, the approximation error $O\left(\frac{N^2}{k}\right)$ dominates while the estimation error remains minimal. As k reaches medium values ($k \approx \frac{T}{e}$), the estimation error peaks while the approximation error begins to decrease. For large k when $k \to T$, the approximation error approaches zero as expert capacity becomes sufficient, while the estimation error decreases but remains non-negligible due to N scaling. The optimal k^* must balance these competing errors.

N **Dependence (Task Complexity):** Both approximation and estimation errors scale with N, but the approximation error grows faster at large k. For large N, larger k values are favored to reduce the approximation error.

d_N **Dependence (Model Complexity):** A high router complexity d_N amplifies the estimation error penalty for large k, favoring $k^* < T$.

m **Dependence (Dataset Size):** The estimation error diminishes with increasing m, reducing its impact at large k. This allows $k = T$ to minimize $E(k, N)$. For large m, the approximation error tends to dominate. For small k, it may gradually turn favoring $k^* < T$.

T **Dependence (Total Experts):** Increasing T increases the estimation error if k is kept constant. To prevent the estimation error from increasing, k should be increased proportionally with T.

Overall, we conclude that (a): Under large dataset regime (m is large), a larger k^* is preferred. (b): Under high router complexity regime (d_N is large), a smaller k^* is preferred to mitigate the increased estimation error. (c): Under increasing task complexity setting, a larger k^* is preferred. (d): When the total number of available experts T is increasing, k needs to be adjusted proportionally to prevent an increase in the estimation error.

5. Conclusion

In this work, we investigated whether conventional sparse activation strategies in SMoE models are optimal for compositional tasks. Through both empirical and theoretical analyses, we showed that the optimal number of activated experts scales with task complexity and depends on training data size and model architecture. Our experiments revealed that increasing the number of experts improves generalization on harder compositional tasks but can degrade performance on simpler ones, highlighting the need for adaptive sparsity. Theoretically, we derived generalization error bounds for SMoE models under compositional settings, identifying a trade-off between approximation and estimation errors. These findings challenge traditional sparse activation assumptions and provide actionable insights for designing SMoE models. Future work will explore dynamic routing strategies to adapt expert activation levels based on real-time task complexity.

References

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

[2] Sarthak Mittal, Yoshua Bengio, and Guillaume Lajoie. Is a modular architecture enough? *Advances in Neural Information Processing Systems*, 35:28747–28760, 2022.

[3] 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, LÃľlio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, ThÃľophile Gervet, Thibaut Lavril, Thomas Wang, TimothÃľe Lacroix, and William El Sayed. Mixtral of experts, 2024.

[4] xAI. Open release of grok-1. https://x.ai/blog/grok-os.

[5] Databricks. Introducing dbrx: A new state-of-the-art open llm. https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm.

[6] 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. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2401.06066.

[7] Qwen. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters", February 2024. URL https://qwenlm.github.io/blog/qwen-moe/.

[8] SambaNova. Samba-coe v0.3: The power of routing ml models at scale. https://sambanova.ai/blog/samba-coe-the-power-of-routing-ml-models-at-scale.

[9] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer-mamba language model, 2024. URL https://arxiv.org/abs/2403.19887.

[10] Simon Schug, Seijin Kobayashi, Yassir Akram, JoÃčo Sacramento, and Razvan Pascanu. Attention as a hypernetwork, 2024. URL https://arxiv.org/abs/2406.05816.

[11] Dingli Yu, Simran Kaur, Arushi Gupta, Jonah Brown-Cohen, Anirudh Goyal, and Sanjeev Arora. Skill-mix: A flexible and expandable family of evaluations for ai models. *arXiv preprint arXiv:2310.17567*, 2023.

[12] Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

[13] Noam Chomsky. *Aspects of the Theory of Syntax*. Number 11. MIT press, 2014.

[14] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N. Siddharth, Brooks Paige, Dana H. Brooks, Jennifer Dy, and Jan-Willem van de Meent. Structured disentangled representations, 2018. URL https://arxiv.org/abs/1804.02086.

[15] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, 31, 2018.

[16] Milton Llera Montero, Casimir JH Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations*, 2020.

[17] Maya Okawa, Ekdeep S Lubana, Robert Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. *Advances in Neural Information Processing Systems*, 36, 2024.

[18] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pages 8489–8510. PMLR, 2023.

[19] Evelina Leivada, Elliot Murphy, and Gary Marcus. Dall· e 2 fails to reliably capture common syntactic processes. *Social Sciences & Humanities Open*, 8(1):100648, 2023.

[20] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[21] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage?, 2023. URL https://arxiv.org/abs/2304.15004.

[22] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36, 2024.

[23] Nayoung Lee, Kartik Sreenivasan, Jason D. Lee, Kangwook Lee, and Dimitris Papailiopoulos. Teaching arithmetic to small transformers, 2023. URL https://arxiv.org/abs/2307.03381.

[24] Simon Schug, Seijin Kobayashi, Yassir Akram, Maciej Wołczyk, Alexandra Proca, Johannes Von Oswald, Razvan Pascanu, João Sacramento, and Angelika Steger. Discovering modular solutions that generalize compositionally. *arXiv preprint arXiv:2312.15001*, 2023.

[25] Sanjeev Arora and Anirudh Goyal. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*, 2023.

[26] Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. Harder tasks need more experts: Dynamic routing in moe models, 2024. URL https://arxiv.org/abs/2403.07652.

[27] Samira Abnar, Harshay Shah, Dan Busbridge, Alaaeldin Mohamed Elnouby Ali, Josh Susskind, and Vimal Thilak. Parameters vs flops: Scaling laws for optimal sparsity for mixture-of-experts language models, 2025. URL https://arxiv.org/abs/2501.12370.

[28] Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards understanding mixture of experts in deep learning. *arXiv preprint arXiv:2208.02813*, 2022.

[29] Huy Nguyen, TrungTin Nguyen, Khai Nguyen, and Nhat Ho. Towards convergence rates for parameter estimation in gaussian-gated mixture of experts, 2023.

[30] Huy Nguyen, TrungTin Nguyen, and Nhat Ho. Demystifying softmax gating function in gaussian mixture of experts, 2023.

[31] Huy Nguyen, Pedram Akbarian, TrungTin Nguyen, and Nhat Ho. A general theory for softmax gating multinomial logistic mixture of experts, 2023.

[32] Huy Nguyen, Pedram Akbarian, Fanqi Yan, and Nhat Ho. Statistical perspective of top-k sparse softmax gating mixture of experts, 2023.

[33] Huy Nguyen, Nhat Ho, and Alessandro Rinaldo. Sigmoid gating is more sample efficient than softmax gating in mixture of experts. *arXiv preprint arXiv:2405.13997*, 2024.

[34] Samy Jelassi, Clara Mohri, David Brandfonbrener, Alex Gu, Nikhil Vyas, Nikhil Anand, David Alvarez-Melis, Yuanzhi Li, Sham M Kakade, and Eran Malach. Mixture of parrots: Experts improve memorization more than reasoning. *arXiv preprint arXiv:2410.19034*, 2024.

[35] Wissam Akretche, Frédéric LeBlanc, and Mario Marchand. Tighter risk bounds for mixtures of experts. *arXiv preprint arXiv:2410.10397*, 2024.

[36] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

[37] Dongyang Fan, Bettina Messmer, and Martin Jaggi. Towards an empirical understanding of moe design choices. *arXiv preprint arXiv:2402.13089*, 2024.

[38] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, SimÃşn Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, ÅĄukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, ÅĄukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David MÃľly, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John

Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe CerÃşn Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[39] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them, 2019.

[40] Ying Jin. Upper bounds on the natarajan dimensions of some function classes, 2023.

[41] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2022.

[42] Ming Zhong, Aston Zhang, Xuewei Wang, Rui Hou, Wenhan Xiong, Chenguang Zhu, Zhengxing Chen, Liang Tan, Chloe Bi, Mike Lewis, et al. Law of the weakest link: Cross capabilities of large language models. *arXiv preprint arXiv:2409.19951*, 2024.

[43] J. Raven, J. Raven, and Competency Motivation Project. *Uses and Abuses of Intelligence: Studies Advancing Spearman and Raven's Quest for Non-arbitrary Metrics*. Royal Fireworks Press, 2008. ISBN 9780955719509. URL https://books.google.com/books?id=UMwxKQAACAAJ.

[44] Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36, 2023.

[45] Balas K Natarajan. On learning sets and functions. *Machine Learning*, 4:67–97, 1989.

[46] Ron Meir and Tong Zhang. *Generalization error bounds for Bayesian mixture algorithms*. Technion, Israel Institute of Technology, 2003.

A. Appendix

A.1. More Related Works

SRAVEN symbolic reasoning test Inspired by Raven Progressive Matrices (RAVEN) Test [43], a human Intelligence Quotient test on abstract reasoning, Schug et al. [10] proposed the symbolic compositional SRAVEN test that requires a model to learn the composition of arbitrarily sampled rules by searching through a large number of possible hypotheses. Similar to RAVEN, each SRAVEN task is a 3×3 grid and the model is asked to query the final panel on the grid given the information from the first 8 panels. Each panel is a vector of length M, where each entry on the vector corresponds to a different rule sampled from $R = 8$ possible rules. Therefore, each task is composed by a finite set of rule combinations, and the prediction will be marked as correct only if the model predict all the entries of the final vector. More detailed description can be found in Section 4 in [10]. In this paper, we trained SMoE-based transformers on SRAVEN task due to its compositional nature and the flexibility on tuning the difficulty of the task.

Skill-Mix Skill-Mix [11] is a novel evaluation method for assessing language models' compositional abilities. It challenges models to generate short text pieces combining random sets of k skills out of N number of linguistic skills within a given topic. Therefore, the test's difficulty increases with k. A Grader model (e.g., GPT-4 [38]) is used to evaluate the generated outputs based on skill application, topic relevance, length, and coherence. In this paper, we tested SMoE-based LLMs on Skill-Mix with varying number of k. More experimental and grading details can be found in Section C in Yu et al. [11].

A.2. Training SMoE-based Transformers on SRAVEN task

A.2.1. Ablation study: Switching Softmax attention to HYLA attention

Proposed by Schug et al. [10], Hypernetwork Linear Attention (HYLA) encourages compositional generalization by reinforcing the hypernetwork perspective. We repeat the same set of experiments by replacing softmax attention with HYLA attention. We have the following interesting findings:

- By comparing Figure 5 and Figure 1, we observe that HYLA does not improve and can even degrade compositional generalization (particularly OOD accuracy) on SMoE models when training on easier compositional tasks (i.e., $M = 2$) compared to using softmax attention. However, as the compositional task becomes more challenging, HYLA significantly outperforms softmax attention, enhancing the compositional generalization of the model across all activation mechanisms.
- As shown in Figure 4 and Figure 5, sparse activation mechanisms like Top-1 and Top-2 remain the worst-performing routing strategies across almost all task difficulty levels, even with HYLA attention. This finding aligns with our main results discussed in Section 3.1.1.
- As illustrated in Figure 2 and Figure 1, under the softmax attention setting, increasing the number of activated experts leads to marginal improvements in both Test and OOD accuracy as the compositional task difficulty increases. In contrast, under HYLA attention, the same increase in activated experts results in dramatic improvements for more challenging compositional tasks, as shown in the last row of Figure 4 and Figure 5. The optimal number of activated experts scales proportionally with task difficulty M, and the performance gap between the best and second-best routing mechanisms becomes significant. We hypothesize that HYLA attention further enhances expert specialization for compositional tasks.

  Overall, HYLA attention coupled with increased expert activation is promising for robust compositional generalization in challenging tasks, while simpler tasks may not benefit significantly from these adjustments.

A.2.2. SRAVEN Experiments Results Details

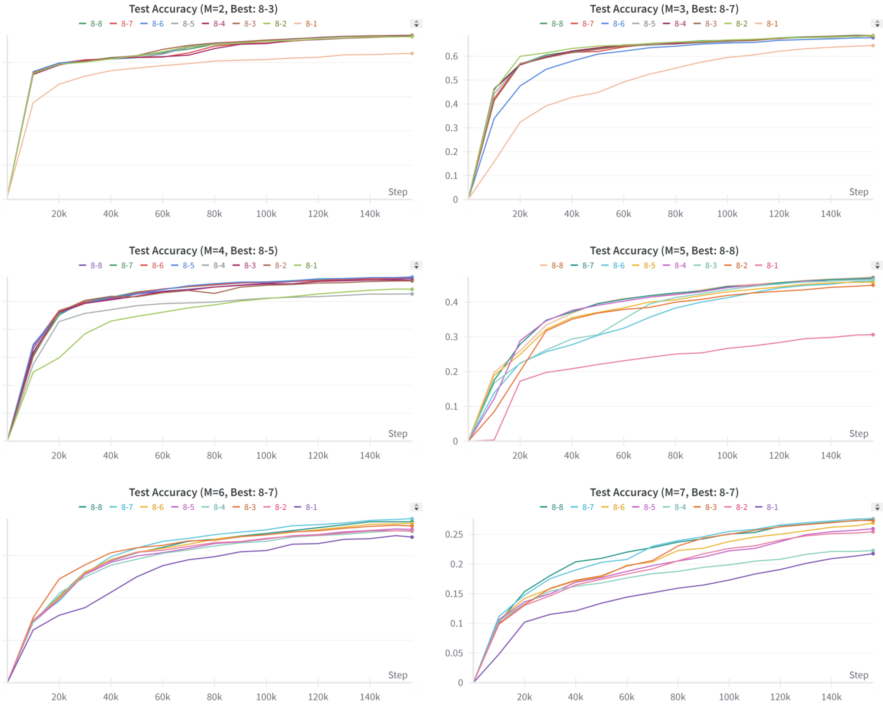

Figure 2: Test Accuracy of training SMoE Transformer with softmax attention.

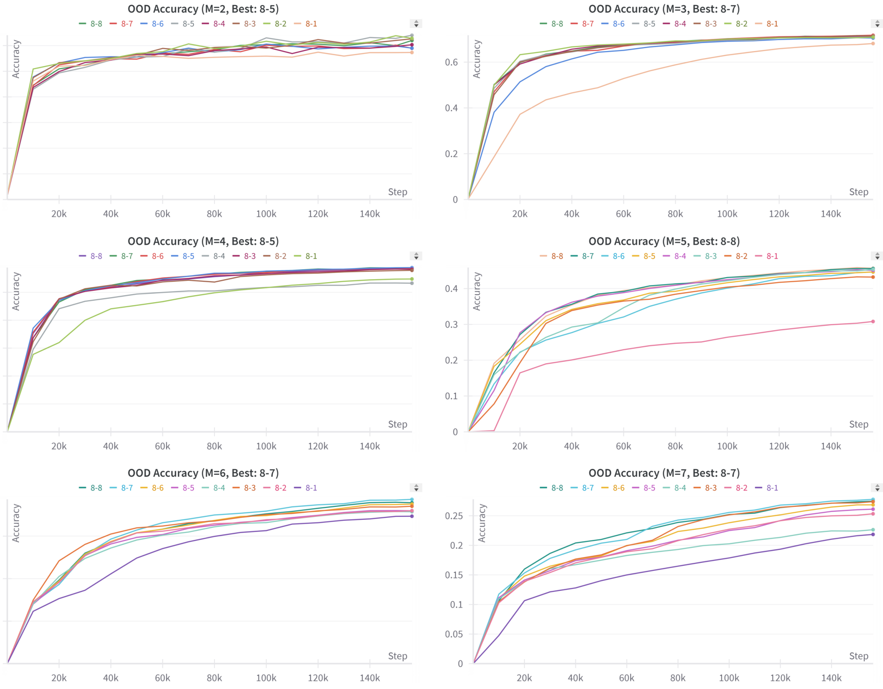

Figure 3: OOD Accuracy of training SMoE Transformer with softmax attention.

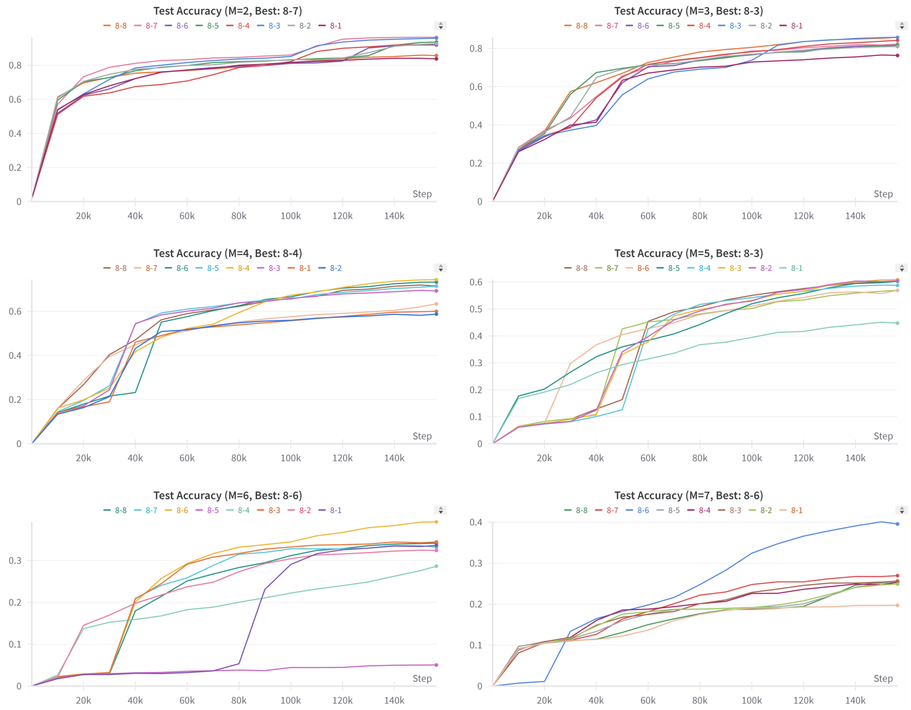

Figure 4: Test Accuracy of training SMoE Transformer with hypernetwork linear attention.

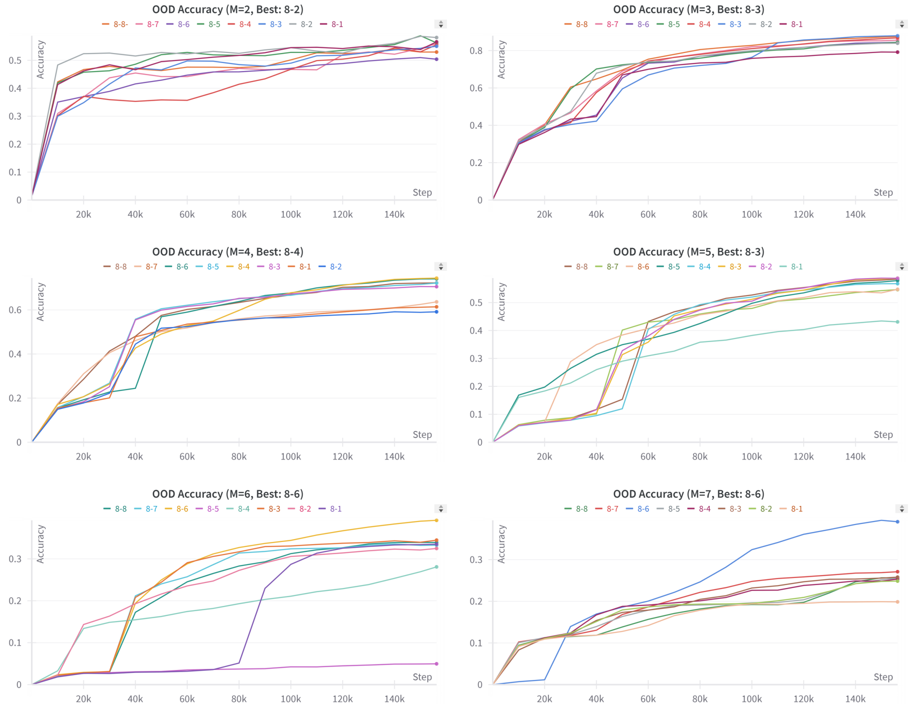

Figure 5: OOD Accuracy of training SMoE Transformer with hypernetwork linear attention.

A.3. Evaluating SMoE-based Large Language Models on Skill-Mix

We copied the grading metrics definitions from Yu et al. [11] as a reference for the readers. Note that the first three grading metrics are very tough and most models will earn very few points when k increases, as these metrics are conditioned on a specific event.

A.3.1. Skill-Mix Grading Metrics Definition

Each generated text can receive up to $k + 3$ points: 1 point for each correctly illustrated skill, 1 point for sticking to the topic, 1 point for coherence / making sense, and 1 point for having at most $k - 1$ sentence. Recall that we grade each generated text three times. In each round of grading, we parse each of the criteria individually from the Grader model's output. For each criterion, we then collect the majority vote among the three grading rounds. The grading metrics are the following:

- *Ratio of Full Marks*: 1 if all $k + 3$ points are earned, and 0 otherwise
- *Ratio of All Skills*: 1 if k points are awarded for the k skills and at least 2 points are awarded for the remaining criteria, and 0 otherwise
- *Skill Fraction*: the fraction of points awarded for the k skills if all 3 points are awarded for the remaining criteria, and 0 otherwise

We then take the maximum value of the metrics among the 3 generations for a given (k skill, 1 topic) combination, and average the maximum value across all the combinations.

A.4. More Skill-Mix Results

EPT	Skill-Mix results for DBRX-132B evaluated by GPT4			
	$k = 1$	$k = 2$	$k = 3$	$k = 4$
1	0.00 ± 0.000	0.00 ± 0.000	0.00 ± 0.000	0.00 ± 0.000
	0.00 ± 0.000	0.00 ± 0.000	0.00 ± 0.000	0.00 ± 0.000
	0.00 ± 0.000	0.00 ± 0.000	0.00 ± 0.000	0.00 ± 0.000
2	0.60 ± 0.245	0.20 ± 0.200	0.00 ± 0.000	0.00 ± 0.000
	1.00 ± 0.000	0.40 ± 0.245	0.00 ± 0.000	0.00 ± 0.000
	0.60 ± 0.245	0.40 ± 0.187	0.33 ± 0.000	0.15 ± 0.100
3	0.80 ± 0.200	0.40 ± 0.245	0.00 ± 0.000	0.00 ± 0.000
	1.00 ± 0.000	0.60 ± 0.245	0.00 ± 0.000	0.00 ± 0.000
	0.80 ± 0.200	0.70 ± 0.122	0.60 ± 0.067	0.35 ± 0.061
4 (default setting)	**1.00** ± **0.000**	0.40 ± 0.245	0.00 ± 0.000	0.00 ± 0.000
	1.00 ± **0.000**	0.40 ± 0.245	0.00 ± 0.000	0.00 ± 0.000
	1.00 ± **0.000**	0.70 ± 0.122	0.60 ± 0.067	0.40 ± 0.061
5	**1.00** ± **0.000**	**0.60** ± **0.245**	0.20 ± 0.200	0.00 ± 0.000
	1.00 ± **0.000**	**0.80** ± **0.200**	0.20 ± 0.200	0.00 ± 0.000
	1.00 ± **0.000**	**0.80** ± **0.122**	0.67 ± 0.105	0.45 ± 0.094
6	0.80 ± 0.200	0.40 ± 0.245	0.20 ± 0.200	0.00 ± 0.000
	0.80 ± 0.200	0.40 ± 0.245	0.20 ± 0.200	0.00 ± 0.000
	0.80 ± 0.200	0.70 ± 0.122	0.60 ± 0.163	0.50 ± 0.000
7	1.00 ± 0.000	0.40 ± 0.245	0.20 ± 0.200	0.00 ± 0.000
	1.00 ± 0.000	0.40 ± 0.245	0.20 ± 0.200	0.00 ± 0.000
	1.00 ± 0.000	0.70 ± 0.122	0.67 ± 0.105	0.65 ± 0.061
8	0.80 ± 0.200	0.20 ± 0.200	0.20 ± 0.200	0.00 ± 0.000
	1.00 ± 0.000	0.20 ± 0.200	0.20 ± 0.200	0.00 ± 0.000
	0.80 ± 0.200	0.60 ± 0.100	0.67 ± 0.105	0.50 ± 0.112
9	0.80 ± 0.200	0.40 ± 0.245	0.20 ± 0.200	0.00 ± 0.000
	1.00 ± 0.000	0.80 ± 0.200	0.40 ± 0.245	0.00 ± 0.000
	0.80 ± 0.200	0.60 ± 0.187	0.67 ± 0.105	0.55 ± 0.050
10	1.00 ± 0.000	0.20 ± 0.200	0.00 ± 0.000	0.00 ± 0.000
	1.00 ± 0.000	0.60 ± 0.245	0.00 ± 0.000	0.00 ± 0.000
	1.00 ± 0.000	0.60 ± 0.100	0.53 ± 0.082	0.40 ± 0.061
11	1.00 ± 0.000	0.60 ± 0.245	0.20 ± 0.200	0.00 ± 0.000
	1.00 ± 0.000	0.60 ± 0.245	0.20 ± 0.200	0.00 ± 0.000
	1.00 ± 0.000	0.80 ± 0.122	0.47 ± 0.170	0.55 ± 0.094
12	0.80 ± 0.200	0.60 ± 0.245	0.00 ± 0.000	0.00 ± 0.000
	1.00 ± 0.000	0.60 ± 0.245	0.20 ± 0.200	0.00 ± 0.000
	0.80 ± 0.200	0.80 ± 0.122	0.40 ± 0.163	0.50 ± 0.057
13	1.00 ± 0.000	0.40 ± 0.245	**0.40** ± **0.245**	0.00 ± 0.000
	1.00 ± 0.000	0.40 ± 0.245	**0.60** ± **0.245**	0.00 ± 0.000
	1.00 ± 0.000	0.60 ± 0.187	**0.67** ± **0.149**	0.50 ± 0.079
14	1.00 ± 0.000	0.60 ± 0.245	0.00 ± 0.000	**0.20** ± **0.200**
	1.00 ± 0.000	0.60 ± 0.245	0.00 ± 0.000	**0.20** ± **0.200**
	1.00 ± 0.000	0.60 ± 0.245	0.53 ± 0.082	**0.70** ± **0.122**
15	1.00 ± 0.000	0.20 ± 0.200	0.00 ± 0.000	0.00 ± 0.000
	1.00 ± 0.000	0.60 ± 0.245	0.00 ± 0.000	0.00 ± 0.000
	1.00 ± 0.000	0.40 ± 0.187	0.53 ± 0.082	0.40 ± 0.061
16	0.80 ± 0.200	0.20 ± 0.200	0.20 ± 0.200	0.00 ± 0.000
	0.80 ± 0.200	0.40 ± 0.245	0.40 ± 0.245	0.00 ± 0.000
	0.80 ± 0.200	0.50 ± 0.158	0.47 ± 0.170	0.25 ± 0.112

Table 2: Skill-Mix Evaluation Results on DBRX-132B [5]. The grading metrics are Ratio of Full Marks/Ratio of All Skills/Skill Fraction as defined in Section A.3.1. ’EPT’ is the abbreviation for ’Number of experts per token’.

A.5. Power-Law Distribution of compositional task difficulties

In this section, we describe the more complex and realistic *Power-Law Distribution* [42] setting of compositional tasks.

A.5.1. Task-specific Variability

We argue that the simplified Approximation Error defined in Definition 4 treats every task pair uniformly, assuming k experts distribute their capacity evenly across all N^2 task combinations. This assumption, while simplifying the analysis, does not account for task-specific variability:

- Some combinations might require significantly more capacity (e.g., harder-to-learn tasks).
- Some combinations might overlap or share features, reducing the need for dedicated experts across all N^2 pairs.

If the approximation error is task-specific, a more sophisticated construction is necessary to reflect the heterogeneity of task demands. For example, one can craft weight combinations based on difficulty or importance:

$$O\left(\frac{\sum_{i,i'} w_{i,i'}}{k}\right),$$

where $w_{i,i'}$ represents the weight of importance or difficulty for task pair $(T_i, T_{i'})$, as recently probed empirically by Zhong et al. [42]. Additionally, $w_{i,i'}$ is distributed according to a *power-law*:

$$w_{i,i'} \propto (D_i + D_{i'})^{-\alpha}, \quad \alpha > 1,$$

where D_i and $D_{i'}$ are the corresponding inverse of the difficulty or importance of compositional tasks T_i and $T_{i'}$. Therefore, high weights are concentrated in a few challenging combinations (smaller $i + i'$), and most $w_{i,i'}$ are near zero for large $i + i'$.
Additionally, the more important/difficult task combinations will dominate, suggesting:

$$\sum_{(i,i') \text{ s.t. } D_i, D_{i'} \text{ are low}} w_{i,i'} \gg \sum_{(i,i') \text{ s.t. } D_i, D_{i'} \text{ are high}} w_{i,i'},$$

This suggests that **Reducing** k focuses capacity on high-weight combinations, minimizing the Approximation Error. If $k = T$, excess experts can dilute the capacity to low-weight combinations, inflating error due to unnecessary model complexity.

Remark 4 (Power-law distribution is more realistic in LLM Evaluation)**.** *The power-law distribution in task importance is more realistic and supported by empirical observations across various studies. Dziri et al.* [22] *highlights how transformers prioritize high-frequency patterns, leaving rare patterns underrepresented, reflecting power-law dynamics. Similarly, Yu et al.* [11] *demonstrates the combinatorial explosion of rare skill requirements, where a few dominant combinations account for most of the performance. Lastly, Chen et al.* [44] *emphasizes skill hierarchies, where foundational skills dominate model capacity, mirroring power-law behavior in skill distributions. Together, these findings hint that activating all experts is practically inefficient for sparse or compositional tasks, as it dilutes capacity across less-relevant combinations.*

B. Complete Proof of main result

We present the proof of our main result as below:

Proof. Following the classical PAC learning theory, the main objective is to show the following probabilistic bound of $\sup_{f \in \mathcal{F}} |L_{\mathcal{S}}(f) - L_{\mathcal{D}}(f)|$. First we use ghost sampling trick: we draw an i.i.d. copies of training samples: $\mathcal{S}' = \{\boldsymbol{x'}_1, \cdots, \boldsymbol{x'}_m\} \overset{i.i.d.}{\sim} \mathcal{D}^m$. Then by Lemma 7, and setting $e^{-\frac{1}{2}\epsilon^2 m} \leq 1/4$, we have

$$\mathbb{P}\left(\sup_{f \in \mathcal{F}} |L_{\mathcal{S}}(f) - L_{\mathcal{D}}(f)| \geq \epsilon\right) \leq 2\mathbb{P}\left(\sup_{f \in \mathcal{F}} |L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f)| \geq \frac{\epsilon}{2}\right) \tag{7}$$

Then our proof proceeds by reformulating the gating function $a(\boldsymbol{x})$. Let us rewrite $a(\boldsymbol{x}) = \mu(\boldsymbol{x}) \odot \nu(\boldsymbol{x})$, where $\odot$ denotes the element-wise multiplication, $\mu(\boldsymbol{x}) : \mathbb{R}^d \to \{0,1\}^T$ produces a binary mask specifying the sparse expert selection ($\|\mu(\boldsymbol{x})\|_0 = k$), and $\nu(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}_+^T$ outputs the normalized weights for selected experts such that $\|a(\boldsymbol{x})\|_1 = 1$. In particular, we note that $\nu(\boldsymbol{x})$ is dependent of the function $\mu(\boldsymbol{x})$. We define $\mathcal{V}|_\mu$ as the class of g induced by $\mathcal{A}$ and $\mu(\boldsymbol{x})$.

Now notice that $\mu(\boldsymbol{x})$ amounts to a multi-class classifier, which maps the input $\boldsymbol{x}$ to one of the sparse patterns $\mathcal{M}(2m)$. Define $\mu_1, \cdots, \mu_\Gamma$ which shatters all the possible sparse patterns produced by $2m$ data samples, then we have

$$\mathbb{P}\left(\sup_{f \in \mathcal{F}} |L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f)| \geq \frac{\epsilon}{2}\right) = \mathbb{P}\left(\sup_{a \in \mathcal{A}} \sup_{\substack{h_j \in \mathcal{H}, \\ \forall j \in [T]}} |L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f)| \geq \frac{\epsilon}{2}\right) \tag{8}$$

$$\leq \mathbb{P}\left(\sup_{\mu \in \{\mu_1, \cdots, \mu_\Gamma\}} \sup_{\substack{h_j \in \mathcal{H}, \forall j \in [T] \\ \nu \in \mathcal{V}|_\mu}} |L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f)| \geq \frac{\epsilon}{2}\right) \tag{9}$$

$$\leq \sum_{t=1}^{\Gamma} \mathbb{P}\left(\sup_{\substack{h_j \in \mathcal{H}, \forall j \in [T] \\ \nu \in \mathcal{V}|_\mu}} |L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f)| \geq \frac{\epsilon}{2} \,\middle|\, \mu = \mu_t\right) \tag{10}$$

$$\leq \Gamma \sup_{\mu^* \in \{\mu_1, \cdots, \mu_\Gamma\}} \mathbb{P}\left(\sup_{\substack{h_j \in \mathcal{H}, \forall j \in [T] \\ v \in \mathcal{V}|_\mu}} |L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f)| \geq \frac{\epsilon}{2} \,\middle|\, \mu = \mu^*\right) \tag{11}$$

$$\leq 2\Gamma \sup_{\mu^* \in \{\mu_1, \cdots, \mu_\Gamma\}} \mathbb{P}\left(\sup_{\substack{h_j \in \mathcal{H}, \forall j \in [T] \\ v \in \mathcal{V}|_\mu}} L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f) \geq \frac{\epsilon}{2} \,\middle|\, \mu = \mu^*\right), \tag{12}$$

where we apply union bound to obtain Eq. 9 and Eq. 12. Next, we do counting to bound Γ. Since μ is essentially a multi-class classifier with $\binom{T}{k}$ classes, by Lemma 6, we plug in the Natarajan dimension of our sparse patterns:

$$\Gamma \leq \binom{T}{k}^{2Nd_N} \cdot (2m)^{Nd_N}. \tag{13}$$

On the other hand, we bound remaining probability term by examining the expectation given a fixed masking function μ. Define function $\phi|_\mu$ conditioned on μ as:

$$\phi|_\mu(\mathcal{S}, \mathcal{S}') = \sup_{\substack{h_j \in \mathcal{H}, \forall j \in [T] \\ \nu \in \mathcal{V}|_\mu}} L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f) \tag{14}$$

By Lemma 11 and Mcdiarmid's inequality, for any μ, we have bound:

$$\mathbb{P}\left(\phi|_\mu(\mathcal{S},\mathcal{S}') \geq \frac{\epsilon}{2}\right) = \mathbb{P}\left(\phi|_\mu(\mathcal{S},\mathcal{S}') - \mathbb{E}\left[\phi|_\mu(\mathcal{S},\mathcal{S}')\right] \geq \frac{\epsilon}{2} - \mathbb{E}\left[\phi|_\mu(\mathcal{S},\mathcal{S}')\right]\right) \tag{15}$$

$$\leq \exp\left(-2m\left(\frac{\epsilon}{2} - \mathbb{E}\left[\phi|_\mu(\mathcal{S},\mathcal{S}')\right]\right)^2\right) \tag{16}$$

Combined with Eq. 12, 13, 16, we state, with probability at least $1-\delta$,

$$\phi|_\mu(\mathcal{S},\mathcal{S}') \leq 2\mathbb{E}\left[\phi|_\mu(\mathcal{S},\mathcal{S}')\right] + 2\sqrt{\frac{\log\left(\binom{T}{k}^{2Nd_N}(2m)^{Nd_N}\right) + \log(2/\delta)}{2m}} \tag{17}$$

$$= 2\mathbb{E}\left[\phi|_\mu(\mathcal{S},\mathcal{S}')\right] + 2\sqrt{\frac{2kNd_N\log T + Nd_N\log(2m) + \log(2/\delta)}{2m}} \tag{18}$$

We conclude the proof by the bounding $\mathbb{E}\left[\phi|_\mu(\mathcal{S},\mathcal{S}')\right] \leq 2C\mathcal{R}_m(\mathcal{H})$ using Lemma 5. □

Lemma 5. *Consider C-Lipschitz loss function: $\ell : \mathcal{Y}\times\mathbb{R}\to\mathbb{R}$, and follow the definition of $\phi|_\mu$ in Eq. 14, we have*

$$\mathbb{E}\left[\phi|_\mu(\mathcal{S},\mathcal{S}')\right] \leq 2C\mathcal{R}_m(\mathcal{H}) \tag{19}$$

Proof. For the sake of notation simplicity, we define a function space conditioned on a masking function μ:

$$\mathcal{F}|_\mu = \left\{ f(\boldsymbol{x}) = \sum_{j=1}^{T}\mu(\boldsymbol{x})_j\nu(\boldsymbol{x})_j h_j(\boldsymbol{x}) : h_1,\cdots,h_T\in\mathcal{H}, \nu\in\mathcal{V}|_\mu \right\} \tag{20}$$

We denote the loss function ℓ composed on $\mathcal{F}|_\mu$ as $\ell\circ\mathcal{F}|_\mu = \{\ell(f(\boldsymbol{x})) : f\in\mathcal{F}|_\mu\}$. By Lemma 8,

$$\mathbb{E}\left[\phi|_u(\mathcal{S},\mathcal{S}')\right] = \mathbb{E}\left[\sup_{\ell_f\in\ell\circ\mathcal{F}|_\mu}\left(\frac{1}{m}\sum_{i=1}^{m}\ell_f(\boldsymbol{x}_i) - \frac{1}{m}\sum_{i=1}^{m}\ell_f(\boldsymbol{x}'_i)\right)\right] \tag{21}$$

$$\leq 2\mathcal{R}_m(\ell\circ\mathcal{F}|_\mu) \tag{22}$$

Since ℓ is Lipschitz function, by Lemma 9, we have

$$\mathcal{R}_m(\ell\circ\mathcal{F}|_\mu) \leq C\mathcal{R}_m(\mathcal{F}|_\mu) \tag{23}$$

Afterwards, we bound $\mathcal{R}_m(\mathcal{F}|_\mu)$ by:

$$\mathbb{E}_{\mathcal{S},\boldsymbol{\sigma}}\left[\frac{1}{m}\sup_{f\in\mathcal{F}|_\mu}\sum_{i=1}^{m}\sigma_i f(\boldsymbol{x}_i)\right] = \mathbb{E}_{\mathcal{S},\boldsymbol{\sigma}}\left[\sup_{\substack{\sup_{h_j\in\mathcal{H},\forall j\in[T]} \\ v\in\mathcal{V}|_\mu}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\sum_{j=1}^{T}\mu(\boldsymbol{x}_i)_j\nu(\boldsymbol{x}_i)_j h_j(\boldsymbol{x}_i)\right] \tag{24}$$

$$\leq \mathbb{E}_{\mathcal{S},\boldsymbol{\sigma}}\left[\sup_{\substack{h_j\in\mathcal{H},\forall j\in[T] \\ \boldsymbol{\lambda}\in\mathbb{R}_+^T,\|\boldsymbol{\lambda}\|_1=1}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\sum_{j=1}^{T}\boldsymbol{\lambda}_j h_j(\boldsymbol{x}_i)\right] \tag{25}$$

$$= \mathcal{R}_m(\mathcal{H}), \tag{26}$$

where we notice that $\sum_{j=1}^{T}\mu(\boldsymbol{x})_j\nu(\boldsymbol{x}_j) = 1$ due to the softmax normalization over the weights of selected experts, then Eq. 25 can be relaxed by supremum over all simplex. The last equation follows from Lemma 10. Now we can conclude the proof by combining Eq. 22, 23, and 26. □

Lemma 6 (Natarajan Lemma [45]). *Given a set of finite data points $\mathcal{S}$ with $|\mathcal{S}| = m$, and a hypothesis space $\mathcal{H}$ of functions $\mathcal{S}\to[k]$ with Natarajan dimension d_N, then the growth function is bounded by:*

$$\tau_{\mathcal{H}}(m) \leq m^{d_N}\cdot k^{2d_N} \tag{27}$$

Proof. See [45]. □

Lemma 7 (Ghost Sampling [41])**.** *Given $\mathcal{S}$ and $\mathcal{S}'$ with $|\mathcal{S}| = |\mathcal{S}'| = m$, we have the following inequality for any hypothesis space $\mathcal{H}$:*

$$\left(1 - 2e^{-\frac{1}{2}\epsilon^2 m}\right) \mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{S}'}(h)| > \epsilon\right] \leq \mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{S}'}(h)| > \frac{\epsilon}{2}\right] \tag{28}$$

Proof. We note that $\mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{D}}(h)| > \epsilon\right] > 0$, then

$$\mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{S}'}(h)| > \frac{\epsilon}{2}\right] \tag{29}$$

$$\geq \mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{S}'}(h)| > \frac{\epsilon}{2} \cap \sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{D}}(h)| > \epsilon\right] \tag{30}$$

$$= \mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{S}'}(h)| > \frac{\epsilon}{2}\right] \times \mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{S}'}(h)| > \frac{\epsilon}{2} \middle| \sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{D}}(h)| > \epsilon\right] \tag{31}$$

Fix the dataset $\mathcal{S}$ for the event on which we are conditioning. Let h^* be any hypothesis for which $|L_{\mathcal{S}}(h) - L_{\mathcal{D}}(h)| > \epsilon$, then:

$$\mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{S}'}(h)| > \frac{\epsilon}{2} \middle| \sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{D}}(h)| > \epsilon\right] \tag{32}$$

$$\geq \mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h^*) - L_{\mathcal{S}'}(h^*)| > \frac{\epsilon}{2} \middle| \sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{D}}(h)| > \epsilon\right] \tag{33}$$

$$\geq \mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}'}(h^*) - L_{\mathcal{D}}(h^*)| \leq \frac{\epsilon}{2} \middle| \sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{D}}(h)| > \epsilon\right] \tag{34}$$

$$\geq 1 - 2e^{-\frac{1}{2}\epsilon^2 m}, \tag{35}$$

where the last inequality follows from Hoeffding's inequality. Following an averaging over $\mathcal{S}$ argument, we conclude that:

$$\left(1 - 2e^{-\frac{1}{2}\epsilon^2 m}\right) \mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{S}'}(h)| > \epsilon\right] \leq \mathbb{P}\left[\sup_{h\in\mathcal{H}} |L_{\mathcal{S}}(h) - L_{\mathcal{S}'}(h)| > \frac{\epsilon}{2}\right] \tag{36}$$

□

Lemma 8. *Given any funtion class $\mathcal{F}$, for any $\mathcal{S}$ and $\mathcal{S}'$ drawn i.i.d. from $\mathcal{D}^m$ with $|\mathcal{S}| = |\mathcal{S}'| = m$, it holds that*

$$\mathbb{E}_{\mathcal{S},\mathcal{S}'}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{m}\sum_{i=1}^{m} f(\boldsymbol{x}_i) - \frac{1}{m}\sum_{i=1}^{m} f(\boldsymbol{x'}_i)\right)\right] \leq 2\mathcal{R}_m(\mathcal{F}) \tag{37}$$

Proof. The proof is concluded by the following derivation:

$$\mathbb{E}\left[\sup_{f\in\mathcal{F}} L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f)\right] = \mathbb{E}_{\mathcal{S}}\left[\mathbb{E}_{\mathcal{S}'}\left[\sup_{f\in\mathcal{F}} |L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f)|\right]\right] \tag{38}$$

$$\leq \mathbb{E}_{\mathcal{S},\mathcal{S}'}\left[\mathbb{E}_{\sigma_i}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{m}\sum_{i=1}^{m}\left(f(z_i) - \frac{1}{m}\sum_{i=1}^{m} f(z_i')\right)\right)\right]\right] \tag{39}$$

$$\leq \mathbb{E}_{\mathcal{S},\mathcal{S}',\sigma_i}\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{m}\sum_{i=1}^{m}\sigma_i f(z_i)\right) + \sup_{f\in\mathcal{F}}\left(\frac{1}{m}\sum_{i=1}^{m} -\sigma_i f(z_i')\right)\right] \tag{40}$$

$$= 2\mathcal{R}_m(\mathcal{F}), \tag{41}$$

where we introduce Radamacher random variables $\sigma_i, i = 1, \cdots, m$ in Eq. 39. □

Lemma 9. *Suppose $\mathcal{H} \subseteq \{h : \mathcal{X} \to \mathcal{Y}\}$ and function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ is a C-Lipschitz function, define define $\ell \circ \mathcal{H} = \{\ell \circ h : \forall h \in \mathcal{H}\}$, then $\mathcal{R}_m(\ell \circ \mathcal{H}) \leq C\mathcal{R}_m(\mathcal{H})$.*

Proof. See Meir and Zhang [46]. □

Lemma 10. *Suppose $\mathcal{H} \subseteq \{h : \mathcal{X} \to \mathcal{Y}\}$, then all functions constructed by convex combinations of $\mathcal{H}$ satisfies:*

$$\mathbb{E}_{\mathcal{S},\boldsymbol{\sigma}}\left[\sup_{\substack{h_j \in \mathcal{H}, \forall j \in [T] \\ \boldsymbol{\lambda} \in \mathbb{R}_+^T, \|\boldsymbol{\lambda}\|_1 = 1}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i \sum_{j=1}^{T} \boldsymbol{\lambda}_j h_j(\boldsymbol{x}_i)\right] = \mathcal{R}_m(\mathcal{H}). \tag{42}$$

Proof.

$$\begin{aligned}
\mathbb{E}_{\mathcal{S},\boldsymbol{\sigma}}\left[\sup_{\substack{h_j \in \mathcal{H}, \forall j \in [T] \\ \boldsymbol{\lambda} \in \mathbb{R}_+^T, \|\boldsymbol{\lambda}\|_1 = 1}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i \sum_{j=1}^{T} \boldsymbol{\lambda}_j h_j(\boldsymbol{x}_i)\right] &= \mathbb{E}_{\mathcal{S},\boldsymbol{\sigma}}\left[\sup_{\substack{h_j \in \mathcal{H}, \\ \forall j \in [T]}} \sup_{\substack{\boldsymbol{\lambda} \in \mathbb{R}_+^T, \\ \|\boldsymbol{\lambda}\|_1 = 1}} \frac{1}{m}\sum_{j=1}^{T} \boldsymbol{\lambda}_j \left(\sum_{i=1}^{m} \sigma_i h_j(\boldsymbol{x}_i)\right)\right] && (43) \\
&= \mathbb{E}_{\mathcal{S},\boldsymbol{\sigma}}\left[\sup_{h_{j^*} \in \mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h_{j^*}(\boldsymbol{x}_i)\right] && (44) \\
&= \mathcal{R}_m(\mathcal{H}), && (45)
\end{aligned}$$

where Eq. 44 uses the fact that $\sum_{j=1}^{T} \boldsymbol{\lambda}_j y_j \leq \max_{j=1,\cdots,T} y_j$ for any convex coefficients $\boldsymbol{\lambda}$. Moreover, the equality is achieved if and only if $\boldsymbol{\lambda}_j = 1$ for $j = \arg\max_{j=1,\cdots,T} y_j$ and $\boldsymbol{\lambda}_j = 0$ otherwise. □

Lemma 11. *For arbitrary loss function $\ell : \mathcal{Y} \times \mathbb{R} \to [0,1]$, hypothesis space $\mathcal{H}$, and any $\mathcal{S}, \mathcal{S}' \subseteq \mathcal{X}$ $|\mathcal{S}| = |\mathcal{S}'| = m$, the function*

$$\phi(\mathcal{S}, \mathcal{S}') = \sup_{h \in \mathcal{H}} L_{\mathcal{S}}(f) - L_{\mathcal{S}'}(f) \tag{46}$$

satisfies that

$$|\phi(\mathcal{S}, \mathcal{S}') - \phi(\widehat{\mathcal{S}}, \widehat{\mathcal{S}}')| \leq \frac{1}{m}, \tag{47}$$

where $\widehat{\mathcal{S}}$ and $\widehat{\mathcal{S}}'$ changes one element either in $\mathcal{S}$ or $\mathcal{S}'$ but not both, i.e., if $\mathcal{S}$ is changed to $\widehat{\mathcal{S}}$, then $\mathcal{S}'$ remains unchanged, and vice versa.

Proof. Since $f(\cdot) \in [0,1]$, then it is obvious that $|\phi(\mathcal{S}, \mathcal{S}') - \phi(\widehat{\mathcal{S}}, \widehat{\mathcal{S}}')| \leq \frac{1}{m}$ since we can flip at most one element. □