Mixture of Hidden-Dimensions: Not All Hidden-States' Dimensions are Needed in Transformer

Yilong Chen [1 2] **Junyuan Shang** [3] **Zhenyu Zhang** [3] **Jiawei Sheng** [1 2] **Tingwen Liu** [1 2]
Shuohuan Wang [3] **Yu Sun** [3] **Hua Wu** [3] **Haifeng Wang** [3]

Abstract

Transformer models encounter inefficiency when scaling hidden dimensions due to the uniform expansion of parameters. When delving into the sparsity of hidden dimensions, we observe that only a small subset of dimensions are highly activated, where some dimensions are commonly activated across tokens, and some others uniquely activated for individual tokens. To leverage this, we propose **MoHD (Mixture of Hidden Dimensions)**, a sparse architecture that combines **shared sub-dimensions** for common features and dynamically routes **specialized sub-dimensions** per token. To address the potential information loss from sparsity, we introduce **activation scaling** and **group fusion mechanisms**. MoHD efficiently expands hidden dimensions with minimal computational increases, outperforming vanilla Transformers in both parameter efficiency and task performance across 10 NLP tasks. MoHD achieves 1.7% higher performance with 50% fewer activated parameters and 3.7% higher performance with 3× total parameters expansion at constant activated parameters cost. MoHD offers a new perspective for scaling the model, showcasing the potential of hidden dimension sparsity.

1. Introduction

Large Language Models (LLMs) (Anthropic, 2023; OpenAI, 2023; Touvron et al., 2023a) have achieved impressive performance in various natural language processing tasks. Recent study (Kaplan et al., 2020) suggests that scaling models by increasing parameters and computational resources can enhance their capabilities with sufficient data. However, the large number of parameters in LLMs often results in high training and inference costs. To address this, flexible model architectures (Jiang et al., 2024b; Cai et al., 2024b;c) are sought to enable parameter scaling while maintaining computational efficiency. In Transformers, the parameters are determined by hidden and intermediate dimensions. Some studies (Qiu et al., 2024; Liu et al., 2024) identify the sparsity of intermediate-dimensional activations, and use them to design adaptive networks, such as MoE (Cai et al., 2024c; Dai et al., 2024; Xue et al., 2024a), or apply pruning (Xia et al., 2023; Chen et al., 2023; Ma et al., 2023b) and local activation mechanisms (Liu et al., 2023a) to boost efficiency.

Although elastic scaling of the intermediate dimension has been studied, scaling hidden dimensions with controllable computational costs remains underexplored (Figure 9). In general, the hidden dimension reflects the embedding size of all tokens, and expanding it can increase the model capacity to capture intricate patterns. However, existing Transformers (Vaswani et al., 2023) treat all token dimensions equally, resulting in significant computational and memory overhead as the hidden dimensions scale.

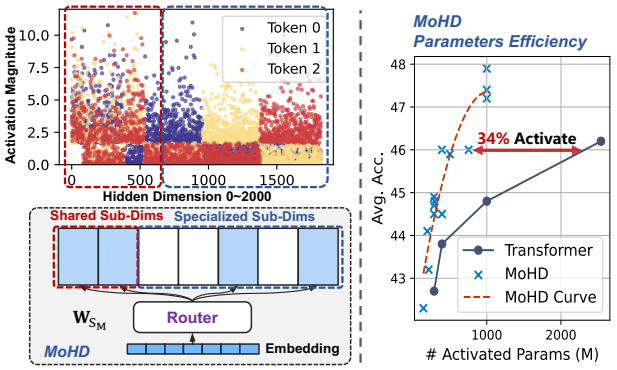


Figure 1. We observe that Transformer hidden states have both token-shared and token-specific activated dimensions. Based on this, we propose MoHD, which combines shared and specialized sub-dimensions for mixed activation. Compared to Transformers, MoHD offers significantly higher parameter efficiency.

[1]Institute of Information Engineering, Chinese Academy of Sciences [2]School of Cyber Security, University of Chinese Academy of Sciences [3]Baidu Inc. Correspondence to: Tingwen Liu <liutingwen@iie.ac.cn>. Project lead: Junyuan Shang <shangjunyuan@baidu.com>.

Proceedings of the 42^{st} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

Considering to limited understanding of hidden dimensions in LLMs, we study the activation magnitudes and find a **significant sparsity**, where 50% of the dimensions contribute 92.54% of the activation magnitudes (Figure 2 left). Among highly activated dimensions, we identify **shared dimensions** consistently activated across tokens, modeling common features, and **specialized dimensions** uniquely activated for individual tokens, capturing higher-level semantics (Figure 3). This inspires an efficient design selectively activating shared and specialized sub-dimensions (Figure 1). Furthermore, we observe a consistent **Activation Flow Pattern** across different blocks (Figure 2 middle, Figure 5), where Attention outputs vary while FFN outputs remain stable, guiding us to design unique and separate sparsity architectures to maintain activation flow integrity.

In this paper, we propose **MoHD (Mixture of Hidden-Dimensions)**, a novel approach that greatly expands hidden dimension capacity through sparse conditional activation, while keeping the activated parameters nearly unchanged. Specifically, MoHD introduces two types of sub-dimensions in each layer of the model's Attention and FFN components: **shared sub-dimensions** that are always activated to capture common dimensional information across different tokens, and **specialized sub-dimensions** that are selectively activated to capture token-specific specialized dimensions. To ensure load balancing, we apply a balancing loss to the specialized sub-dimensions. **An activation scaling mechanism and a grouped fusion mechanism** are introduced to mitigate information loss from dimensional downsampling and maintain the activation flow. In this way, MoHD can be used to scale the model's hidden dimension without increasing the number of parameters, or to significantly reduce the active hidden dimension to lower the computational cost.

To demonstrate the effectiveness of MoHD, we pretrain Vanilla Transformers with 355M, 495M, and 1.13B parameters based on LLaMA's architecture (Touvron et al., 2023b), and their MoHD versions in both hidden dimension compression and expansion settings with scaling factors: 50%, 75%, 2×, 3×, and 4×. We evaluated these models on 10 NLP tasks, showing the advantages of the MoHD architecture. Results indicate that MoHD consistently outperforms Vanilla and Mixture of Experts Transformers with the same activated parameters across all model sizes. In the compression setting, MoHD reduces activated parameters by 50%, retaining 99% of original performance. In the expansion setting, MoHD keeps activated parameters constant while expanding hidden dimensions 4×, achieving up to an 8.37% relative performance improvement. Notably, MoHD-355M outperformed LLaMA2-355M and even matched LLaMA2-1.13B's performance, while reducing activated parameters to 28.9% of LLaMA's. To explore the impact of increasing hidden dimensions, we conducted detailed analyses on MoHD's routing mechanism and sparsification phenomenon. MoHD is the first method to introduce sparse mixture activation for expanding hidden dimensions, offering a new approach to designing efficient architectures.

2. Observation

This section presents key findings for the design of MoHD. In Section 2.1, we observe long-tail sparsity in hidden dimension activations. Section 2.2 examines activation flow in Transformers. In Section 2.3, we identify shared continuous and unique discrete high activation behaviors across tokens, guiding the design of hidden dimension sparsification.

2.1. Sparsity in Tokens' Hidden Dimension

We analyze activation magnitudes of 4096 hidden dimensions in LLaMA2-7B (Figure 2). The left panel reveals the **long-tail sparsity**, consistent with observations in Liu et al. (2024): in the 16th layer Attention input, the top 1000 dimensions account for 71.96% of total magnitude. The middle panel highlights **functional divergence between Transformer's components**: Attention shows higher, fluctuating activations, whereas FFN maintains lower, stable activations. This contrast underscores the need for differentiated activation designs. For details, see the Appendix F.1.

2.2. Activation Flow in Transformer

We also investigate activation magnitude variations within a single Transformer block, as shown in Figure 10. *Consistent **Activation Flow** patterns were observed across blocks.* The activation magnitude variations in each layer follow a similar flow pattern: **The Attention module compresses input activations** to 58% through projections, demonstrating its ability to suppress irrelevant information. **FFN selects and maintains the stability of the activation flow magnitude based on the Attention's output**. Residual connections regulate activation changes by restoring compressed magnitudes. This observation inspires us to maintain the activation flow to reduce information loss. More in Appendix F.2.

2.3. Continuous High Activation

We investigate the temporal correlation of activation sparsity ratio by analyzing high activation values across consecutive tokens and identifying indices repeatedly activated by multiple tokens. **A clear correlation in activations is observed over consecutive tokens.** Figure 2 right shows the number of commonly activated dimensions across 2 to 9 consecutive tokens, with the x-axis representing the threshold for high activation. Using the top 20% activation values as the threshold, 2672 dimensions are commonly activated across 2 tokens, and 673 dimensions remain active across 9 tokens. Figure 3 further illustrates correlated activation

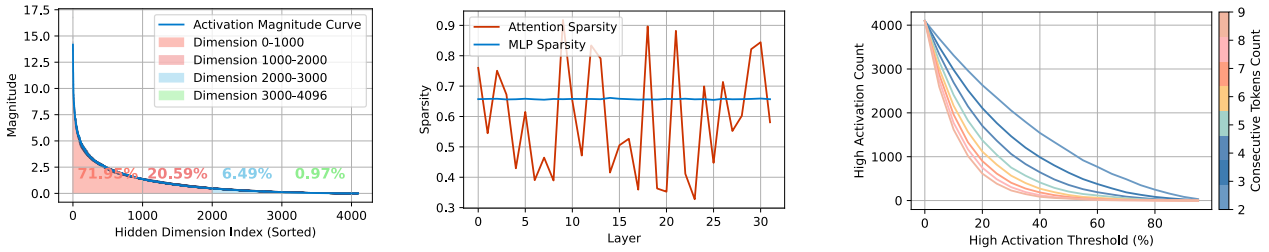


Figure 2. **Left:** Activation magnitudes sorted in descending order with the percentage representing the cumulative activation sum. **Middle:** Sparsity of hidden dimension activations in Attention and FFN outputs across layers. **Right:** Number of shared activation dimensions at varying activation magnitude thresholds, with curves showing the count for consecutive tokens ranges from 2 (blue) to 20 (red).

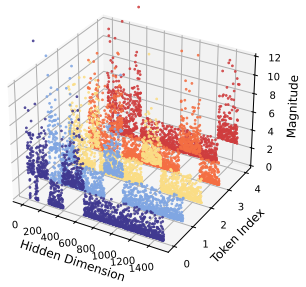


Figure 3. Activation patterns of 4,096 hidden dimensions, clustered and reordered across five tokens in the 16th layer of LLaMA2-7B.

patterns over 5 tokens, where 4096 hidden dimensions are clustered based on their activation patterns. About 400 dimensions are commonly activated across all 5 tokens, while roughly 200 dimensions are uniquely activated in each token. This indicates that **each token's activations contain both shared sub-dimensions activated across tokens and token-specific sub-dimensions.** Shared activations capture similarities, while unique activations reflect differences. These observations inspire the shared-specialized activation mechanism in MoHD's design.

3. Mixture of Hidden Dimensions (MoHD)

In this section, we propose the Mixture of Hidden Dimensions (MoHD) architecture to scale the hidden dimension without increasing activated parameters. The workflow is illustrated in Figure 4. Based on observations from Section 2.12.3, we introduce the Shared and Specialized Sub-Dimension Mixed Activation mechanism in Section 3.1. We also apply sparsified components to Attention and FFN blocks (Appendix A) and address information degradation with activation scaling and grouped fusion mechanisms in Section 3.2. In Section 3.4, we discuss a balance loss and present the implementation in Section 3.5.

3.1. Mixture of Sub-Dimensions Activation

Let $X \in \mathbb{R}^{n\times d}$ denote embeddings of n tokens and $x \in \mathbb{R}^{1\times d}$ denote a single token embedding. Given activation sparsity ratio δ, **MoHD dynamically activates subset $S \subseteq [d]$ of parameters in the weight matrix $\mathrm{W} \in \mathbb{R}^{\mathrm{d}\times\mathrm{d}'}$ through hidden dimension sparsity.** Inspired by observation in Section 2.1, We partition W into N sub-dimensions of size d_e such that $Nd_e = d$, structured as $\mathrm{W} = [\mathrm{W}_1, \mathrm{W}_2, ..., \mathrm{W}_\mathrm{N}]$ where $\mathrm{W}_\mathrm{i} = \mathrm{W}[(\mathrm{i}-1)\mathrm{d}_\mathrm{e} : \mathrm{id}_\mathrm{e}]$. The routing gate $g = \mathrm{Gate}(x, \delta, N)$ selects top-K sub-dimensions via:

$$s_i = \mathrm{Softmax}_i\left(x^\top \phi_i^l\right), g_i = \begin{cases} s_i, & s_i \in \mathrm{Topk}(\{s_j\}_{j=1}^N, K), \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where s_i denotes token-to-subspace affinity, ϕ_i^l is the centroid of i -th sub-dimension at layer l, and $K = \lfloor \delta N \rfloor$. The routing weights g_i enable dynamic amplification/suppression of sub-dimension representations during optimization. Activated sub-dimension outputs $\{y_s\}$ are concatenated to produce the final d-dimensional output.

$$y_s = \Big\|_{i=1}^N g_i \,\mathrm{Dim}_i(x_s) = \mathrm{W_S x_s}. \tag{2}$$

The gated concatenation operator $\Big\|_{i=1}^N g_i \mathrm{Dim}_i(x_s)$ dynamically selects sub-dimensions through sparse activation (sparsity δ). Only $K = \lfloor \delta N \rfloor$ sub-dimensions receive non-zero weights g_i, inducing sparsity in both the token embedding x and weight matrix W. This produces output $y_s \in \mathbb{R}^d$ while reducing activated parameters in $\mathrm{W_S}$ to $\delta \cdot \|\mathrm{W}\|_0$.

3.2. Activation Flow Maintenance

In Section 3.1, our sparse activation mechanism generates dimension-sparsified outputs y_s to reduce computation, but risks *information degradation due to softmax-induced weight skewing and suppression of low-weight dimensions during parallel concatenation.* To preserve consistent **Activation Flow** (Section 2.2), we first introduce **Sub-dimension Scaling** with $\alpha = \sum g_i N$, ensuring stable activation magnitudes across dimensions. A **Grouped Fusion Layer** then projects y_s back to the original d-dimensional space using Monarch matrices (Dao et al., 2022; Chen et al., 2024a), structured as $\mathrm{M} = \sum_{\mathrm{i,j}}^{\mathrm{d/r}} \mathrm{m_{i,j}}$ with a receptive field

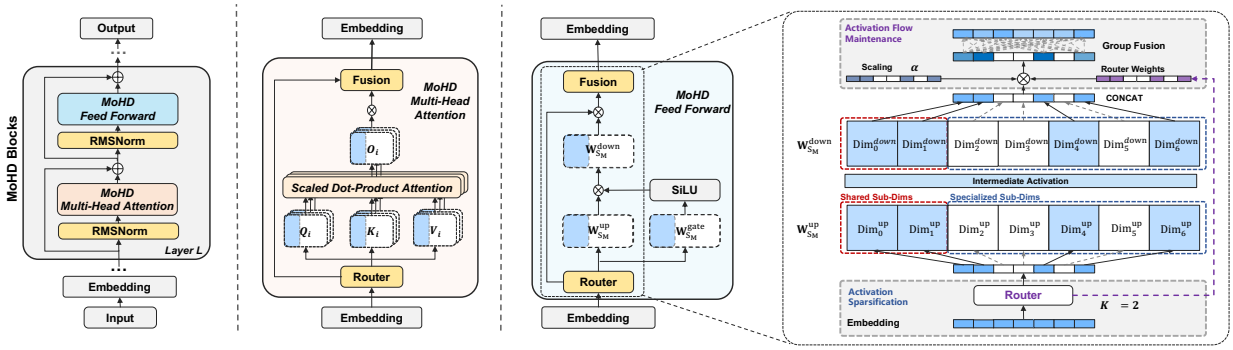


Figure 4. An illustration of MoHD: A single MoHD Block mirrors the structure of a LLaMA Block, consisting of two main components: MoHD Attention and MoHD FFN, each with pre-norm and residual connections. In both components, matrices are selectively activated based on dimensions chosen by the Router. The Router weights shared dimensions and selects a few sparsely activated ones for each token. The resulting outputs from these matrices are weighted, concatenated, and mapped back to the original dimensions.

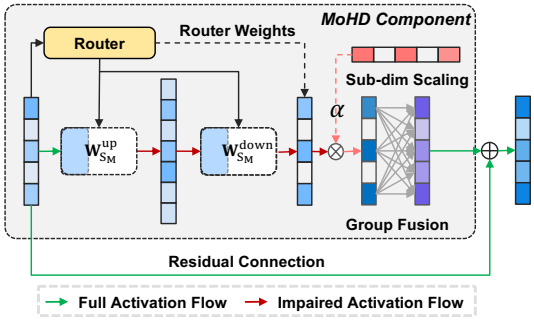


Figure 5. Maintaining Activation Flow: Sparse activations can result in information loss. The scaling factor adjusts the activations to restore the original magnitude, and the grouping and fusion mechanism restores activations to their original dimensionality.

r, reducing the mapping complexity to $O(d/r)$. **Residual Connections** further preserve critical signal flow by bypassing suppressed dimensions. As shown in Figure 5, these components maintain δ-level parameter sparsity while mitigating information loss and computational overhead:

$$\mathrm{My_s} = \begin{bmatrix} m_{1,1} & \cdots & m_{1,d/r} \\ \vdots & \ddots & \vdots \\ m_{d/r,1} & \cdots & m_{d/r,d/r} \end{bmatrix} \otimes \mathrm{y_s}, \tag{3}$$

where the Monarch matrix M enables efficient grouping and transformation, reconstructing information across the original hidden dimensions while maintaining computational efficiency. In summary, the forwarding process for a single MoHD module can be formally represented as follows:

$$y = \mathrm{M}\alpha\|_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{g_i}\,\mathrm{Dim_i}(\mathrm{x_s}), \quad \mathrm{g}, \mathrm{x_s} = \mathrm{Gate}(\mathrm{x}, \delta, \mathrm{N}). \tag{4}$$

3.3. Mixed Activated Sub-Dimensions

As discussed in Section 2.3, *a portion of the hidden dimensions is consistently activated, capturing shared features, while another portion is selectively activated, likely encoding token-specific features.* Motivated by this, we propose a structured routing mechanism that decouples shared and specialized feature encoding. Specifically, we partition each layer's hidden space into two complementary subspaces (Fig. 4): **Shared Sub-Dimensions** (φ portion) maintain constant activation to capture cross-token common features. **Specialized Sub-Dimensions** ($\delta - \varphi$ portion) employs dynamic routing via $\mathrm{Topk}((\delta - \varphi)N)$ selection, encoding fine-grained contextual patterns.

$$s_i = \mathrm{Softmax}_i\left(x^\top \phi_i^l\right),$$
$$g_i = \begin{cases} s_i, & s_i \in \{s_j \mid 1 \leqslant j \leqslant \varphi N\}, \\ s_i, & s_i \in \mathrm{Topk}\left(\{s_j \mid \varphi N \leqslant j \leqslant N\}, (\delta - \varphi)N\right), \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The routing mechanism ensures that **Shared Sub-Dimensions are consistently activated for all tokens, consolidating common information**, while **Specialized Sub-Dimensions are encouraged to diversify.** All sub-dimensions are weighted to allow selective amplification/-suppression of representations during optimization.

3.4. Sub-Dimension Load Balance

Research on conditional computation (Zhou et al., 2022; Jiang et al., 2024b; Dai et al., 2024) has shown that automatically learned routing strategies can often lead to load imbalance issues, where the model tends to select only a few sub-dimensions, leaving others underutilized and insufficiently trained. *To distribute tokens more evenly among different sub-dimensions and smooth out the router score distribution*, we incorporate **Sub-Dimension Load Balance** Loss (Dai et al., 2024): Define β is a scaling factor and $\mathbb{1}_{\{\mathrm{argmax}(g_s)=i\}}$ is an indicator function that returns 1 if the i-th sub-dimension has the highest gating score for the

s-th sequence position and 0 otherwise.

$$\mathbb{L}_{\text{B}} = \beta \sum_{i=1}^{N} \frac{g_i}{\sum_{j=1}^{N} g_j} \cdot \frac{\sum_{s \in S} \mathbb{1}_{\{\text{argmax}(g_s)=i\}}}{N}. \quad (6)$$

The term $\frac{g_i}{\sum_{j=1}^{N} g_j}$ represents the normalized gating score for sub-dimension i, ensuring that the contributions of each sub-dimension are proportional to their selection frequency.

3.5. Implementation

MoHD sparsifies hidden dimensions in Transformer blocks by synchronizing sparsification across up-projection and down-projection matrices, as well as the input x. We define Hidden dimension sparsity and formulate sparsely FFN and Attention mechanisms in Appendix A. Based on these mechanisms, we construct **MoHD BLOCK** using MoHD MHA and MoHD FFN components (Details in Appendix. G). Residual connections are applied to mitigate information loss, and LayerNorm layers are placed before both MHA and FFN inputs. The forward of the block is defined as:

$$\text{BLOCK}_{\text{MoHD}}(x) = \text{FFN}_{\text{MoHD}}(\text{MHA}_{\text{MoHD}}(x) + x) + x. \quad (7)$$

The final training objective combines the cross-entropy loss $\mathbb{L}_{\text{CE}}$ for language modeling with the load balance loss $\mathbb{L}_{\text{B}}$:

$$\mathbb{L} = \mathbb{L}_{\text{CE}} + \mathbb{L}_{\text{B}}. \quad (8)$$

4. Experiments

4.1. Experimental Setup

Data. To pretrain MoHD models and baseline models, we employ the RedPajama (TogetherAI, 2023), which parallels the LLaMA training data across seven domains: CommonCrawl, C4, GitHub, Wikipedia, Books, ArXiv, and StackExchange. This dataset comprises a validation set with 2 million tokens, a training set containing 50 billion tokens.

Training. Our experimental framework utilizes the Sheared-LLaMA codebase (Xia et al., 2023) implemented on the Composer package (Team, 2021), and is executed on 8 NVIDIA A100 GPUs (80GB). The models are trained with a sequence length of 4096, employing a global batch size of 256. MoHD models are trained for 50000 steps (50B token budget). The learning rates were set at 3e-4 for all parameters. The baselines and all MoHD models follow the same training setup, starting from random initialization and training on the same amount of data.

Evaluation. We employed the lm-evaluation-harness (Gao et al., 2021) to evaluate our models. For common sense and reading comprehension tasks, we report 0-shot accuracy results for SciQ (Welbl et al., 2017), PIQA (Bisk et al., 2020), WinoGrande (WG) (Sakaguchi et al., 2020), ARC Easy(ARC-E) (Clark et al., 2018b), and 10-shot HellaSwag (Hella.) (Zellers et al., 2019), alongside 25-shot accuracy for ARC Challenge (ARC-C) (Clark et al., 2018a). In the assessments of continued QA and text understanding, we report 0-shot accuracy for LogiQA (Liu et al., 2020), 32-shot BoolQ (Clark et al., 2019), and 0-shot LAMBADA (Lam.) (Paperno et al., 2016). All reported results are calculated with the mean and stderr of multiple experiments.

Baseline. Following the architecture of LLaMA2, we constructed models at three parameter scales: 355M, 495M, and 1.13B, with hidden dimensions of 1024, 1536, and 2048, as shown in Table 7. For each parameter scale, we develop three variants: Vanilla Transformers (LLaMA architecture) and MoHD-based models. The flexible MoHD architecture allows for compressing activated parameters without changing the total parameter count or expanding the model parameters while retaining the activation size. We experiment with five hidden dimension scaling factors—$0.5\times$, $0.75\times$, $2\times$, $3\times$, and $4\times$—to showcase MoHD's ability to reduce activation and enhance model capacity. All models are the same initialized and pre-trained on 50 billion tokens.

4.2. Result

Capability in Compression. Table 1 (blue rows) demonstrates the capabilities of MoHD on the 355M, 495M, and 1B versions of LLaMA2 with 50% and 75% hidden dimensions activated. Results show that MoHD maintains or even improves performance with partial activation. At 355M, **MoHD with 50% activation incurs only a 0.4% performance loss compared to the baseline**, highlighting hidden dimension sparsity. Furthermore, **MoHD with 75% activation outperforms the fully activated baseline**, achieving gains of 0.5%, 1%, and 1.8% for the 355M, 495M, and 1B models, respectively. **Performance gains increase with model size**: MoHD 50% achieves relative improvements of -0.4%, +0.3%, and +1.7% for the 355M, 495M, and 1B models, showing potential for larger-scale applications. Although compressing activations to 50% slightly reduces Commonsense metric scores, MoHD retains strong language modeling capabilities under low activation settings.

Capability in Expansion. Table 1 (pink rows) shows MoHD's scalability with $2\times$, $3\times$, and $4\times$ hidden dimension expansion. **MoHD achieves performance comparable to models with equivalent parameters while using fewer activated parameters.** For instance, MoHD $2\times$ (355M) exceeds the baseline by 2.2%, outperforming LLaMA2-495M and LLaMA2-1.13B. As the scale increases, the benefits of MoHD become more pronounced, yielding improvements of 2.2%, 0.7%, and 3% for the 355M, 495M, and 1.13M parameter models, respectively. **Tripling the hidden dimension yields optimal results**: MoHD ×3-1.48B achieves a 2.2% improvement over the baseline and 1.5%

Table 1. Comprehensively evaluate the basic capabilities of models with different activated parameters. In particular, MoHD 50%-355M represents a model with 355M total parameters using MoHD to compress 50% hidden dimensions. Green and Red values indicate metrics that exceed or fall below the baseline, respectively. **# Activate** refers to all activated parameters excluding the Embedding layers.

Model-Params	# Activate	Commonsense & Reading Comprehension						Continued		LM	Knowledge	Avg.
		SciQ	PIQA	WG	ARC-E	ARC-C	Hella.	LogiQA	BoolQ	Lam.	MMLU	
LLaMA2-355M	289M	74.0	65.2	50.5	44.7	20.1	31.1	19.5	59.7	36.6	25.2	42.7
MoHD 50%-355M	145M	74.0	65.6	50.4	43.9	19.7	30.7	20.6	54.7	37.7	25.6	42.3
MoHD 75%-355M	217M	75.3	65.6	50.9	44.7	20.9	31.1	22.3	55.8	38.9	26.2	43.2
MoHD ×2 -710M	289M	76.6	67.5	49.8	47.7	23.0	33.4	20.7	60.5	43.3	26.5	44.9
MoHD ×3 -1.06B	289M	77.1	67.8	51.1	47.6	21.8	33.9	20.6	55.8	43.6	25.6	44.5
MoHD ×4 -1.42B	289M	77.6	67.9	49.1	47.0	23.3	33.9	22.1	57.5	44.3	24.7	44.7
LLaMA2-495M	396M	75.4	66.5	51.3	45.5	19.9	32.0	21.7	60.5	38.9	25.8	43.8
MoHD 50%-495M	198M	76.9	67.1	52.7	46.4	20.1	32.3	21.5	57.0	40.7	26.2	44.1
MoHD 75%-495M	297M	76.4	67.3	50.6	45.8	21.1	33.0	23.7	61.8	41.7	26.2	44.8
MoHD ×2 -989M	396M	77.1	67.8	51.1	47.6	21.8	33.9	20.6	55.8	43.6	25.6	44.5
MoHD ×3 -1.48B	396M	77.0	69.0	51.1	48.8	23.6	35.6	22.0	58.6	48.2	26.1	46.0
MoHD ×4 -1.98B	396M	79.1	67.4	49.8	49.1	22.0	35.2	20.7	60.9	47.4	26.1	45.8
LLaMA2-1.13B	1B	81.0	68.1	51.8	49.3	23.2	35.0	21.7	47.0	38.9	26.4	44.2
MoHD 50%-1.13B	503M	78.9	67.8	50.1	48.7	21.2	35.2	21.5	61.1	48.8	25.5	45.9
MoHD 75%-1.13B	755M	80.3	69.3	52.3	50.8	24.5	36.1	22.3	51.2	48.4	25.0	46.0
MoHD ×2 -2.27B	1B	81.2	70.9	54.1	53.0	24.6	38.3	22.4	50.5	52.1	25.5	47.2
MoHD ×3 -3.41B	1B	83.6	69.8	53.1	51.9	25.4	38.3	21.0	56.5	53.0	26.6	47.9
MoHD ×4 -4.55B	1B	82.4	70.0	52.8	51.6	23.4	38.0	23.4	54.9	50.7	26.6	47.4
LLaMA2-2.7B	2.54B	82.5	70.8	56.3	54.4	27.8	39.3	23.5	44.4	37.7	25.3	46.2

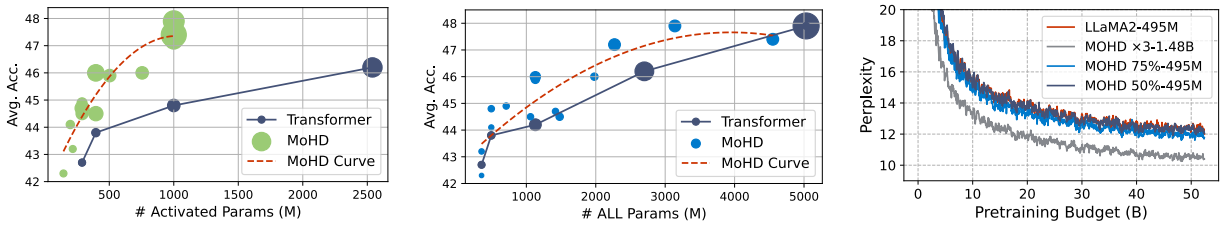


Figure 6. **Left:** Average score with activated parameters, with point size representing total parameters. **Middle:** Average score with total parameters, with point size representing activated parameters. **Right:** Perplexity curves for models pre-trained on 50B tokens.

over MoHD ×2. MoHD effectively expands hidden dimensions and leverages sparsity to boost performance.

Parameters Efficiency. Figure 6 (Left) shows that **MoHD achieves a high parameter efficiency.** On the 400M and 1B scales, MoHD delivers 2.2% and 3% improvements, respectively, over the baseline with less than 50% activation. Performance gains grow with activated parameters, underscoring MoHD's scalability at larger scales. Figure 6 Middle highlights MoHD's advantage with increasing total parameters: at smaller scales, MoHD matches baseline performance with fewer activations, while at larger scales, MoHD achieves greater gains under the same total parameter count by leveraging hidden dimension redundancy.

Training Stability. Figure 6 (Right) visualizes evaluation perplexity during pretraining on 50B tokens for LLaMA2-495M, MoHD ×3-1.48B, MoHD 75%-495M, and MoHD 50%-495M. **MoHD shows training stability and btter efficiency**, with smoother and lower perplexity curves compared to LLaMA2-495M. MoHD ×3-1.48B shows the greatest improvements, while MoHD 75%-495M and MoHD 50%-495M maintain strong training performance, demonstrating robust learning even with partial parameter activation. Overall, MoHD effectively expands hidden dimensions, improving learning capabilities during training.

Compare to MoE. As Sparsely-activated Transformer in a different dimension, we compare MoHD and MoE. DeepSeek MoE (Dai et al., 2024) models are trained from scratch at 355M as the baseline [1]. Even with 75% sparsity, MoHD outperform the baseline while MoE casue perfor-

[1]using Multi-head Attention for fairness. Settings in Table 6.

Table 2. Performance comparison of DeepSeek-V2 MoE (Dai et al., 2024) and MoHD at 355M scale (50B training budgets).

Method	SciQ	PIQA	Hella.	ARC-E	LogiQA	Lam.	Avg. ↑
L.-355M	74.0	65.2	31.1	44.7	19.5	36.6	45.2
MoE 75%	73.7	64.4	30.6	43.1	19.1	35.0	44.3 (-0.9)
MoHD 75%	75.3	65.5	31.1	44.7	22.3	38.9	46.3 (+1.1)
MoE ×2	75.1	65.2	31.9	45.2	20.7	38.0	46.0 (+0.8)
MoHD ×2	76.6	67.5	33.4	47.7	20.7	43.3	48.2 (+3.0)
MoE ×3	76.2	65.8	32.2	46.2	20.1	38.8	46.5 (+1.3)
MoHD ×3	77.1	67.8	33.9	47.6	20.6	43.6	48.4 (+3.2)

mance loss. MoHD outperforms MoE by an average of 2% in both compression and expansion settings. **By expanding the hidden dimension, MoHD enhances the representation ability of each token, demonstrating higher parameter efficiency than MoE.** More details in Appendix D.

4.3. Ablation Studies

To evaluate the importance of each method in Section 3 within MoHD, we conducted detailed ablation experiments. In Table 3, we compare the ablation results of MoHD ×2 with 710M parameters to the baseline with the same activation under zero-shot pretraining on 10B tokens, based on Eval PPL. The specific analysis is as follows:

Mixed Activation Sub-Dimension Ablation. We ablated the Mixed Activation Sub-Dimension method by using fully specialized sub-dimensions without shared ones, leading to a 0.83 increase in PPL, indicating a negative impact on performance. As discussed in Section 2.3, shared activation dimensions should be activated together in a mixed activation mode. Figure 8a shows that this mode significantly outperforms fully sparse activation, highlighting its suitability for hidden dimensions patterns of the Transformer.

Balance Loss Ablation. The balanced loss effectively enhances MoHD (0.16 improvement). It mitigates the risk of routing collapse, ensuring that most sub-dimensions are utilized more evenly. This increases the efficiency of sub-dimension utilization and improves the overall efficiency.

Flow Maintenance Ablation. Ablation experiments highlight that maintaining activation flow is crucial for MoHD. As shown in Table 3, removing Sub-dimension Scaling results in a 1.16 performance drop, as the model loses critical information after sparsifying the hidden dimension. The Group Fusion Layer offers an additional 0.22 performance gain without significantly increasing parameters, improving dimension utilization while preserving information integrity.

4.4. Analysis

Decoupled MoHD Setting. To explore MoHD, we built three models: one sparsifying Attention, one for FFN, and one for both, as shown in Table 4 and Figure 11. All models were trained on 10B tokens. **Sparsifying FFN showed more redundancy**, leading to a minimal or even improved performance. We observe a -0.30 PPL reduction at 50% sparsity, likely due to the reduced redundant activations that mitigate overfitting. **Sparsifying Attention resulted in a greater performance drop**, with a +1.04 PPL increase at 50% sparsity, indicating its sensitivity to sparsification. Finally, **joint sparsification of Attention and FFN achieved the best parameter efficiency**, with the 50%ATTN-50%FFN model reaching a PPL of 12.05 using only 145M activated parameters, outperforming both 50%FFN and 50%ATTN configurations. More detailed analysis in Appendix H.1.

Table 3. Eval Perplexity with ablation on MoHD (10B training budgets). "w.o." indicates the method was ablated.

Method	Perplexity ↓
MoHD ×2 -710M	10.25
w/o Mixed Activated Sub-Dimensions	11.08 (+0.83)
w/o Balance Loss	10.41 (+0.16)
w/o Group Fusion Layer	10.47 (+0.22)
w/o Sub-Dimension Scaling	11.41 (+1.16)
LLaMA2-355M	11.61(+1.36)

Table 4. Eval Perplexity in the MoHD setting is performed for the Attention or FFN of LLaMA2-355M (10B training budgets).

Method	# Activate	Perplexity ↓
LLaMA2-355M	289M	11.61
MoHD-100%ATTN-100%FFN	289M	11.43 (-0.18)
MoHD-100%ATTN-50%FFN	195M	11.31 (-0.30)
MoHD-50%ATTN-100%FFN	239M	12.25 (+0.64)
MoHD-50%ATTN-50%FFN	145M	12.05 (+0.44)
MoHD-100%ATTN-25%FFN	147M	12.24 (+0.63)
MoHD-25%ATTN-100%FFN	213M	14.31 (+2.70)
MoHD-25%ATTN-25%FFN	72M	13.20 (+1.59)

Router Probability. To observe sub-dimension selection in MoHD, we visualize the attention and FFN router weight distributions at the 5th layer across five data domains in Figure 7. Each weight represents the average selection probability across 4096 tokens. The sub-dimensions show specialization across domains: for example, Attention Subdim 5 is crucial for code-related data, with higher probabilities in GitHub and StackExchange. In contrast, Subdim 3 is more relevant to commonsense knowledge, with higher probabilities in Wiki, CC, and ArXiv. In the FFN router, Subdim 4 specializes in code tasks, while Subdim 3 focuses on commonsense knowledge. This specialization validates MoHD's effectiveness in allocating sub-dimensions based

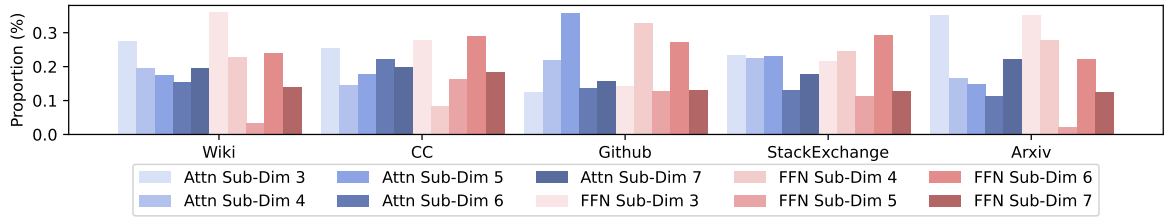
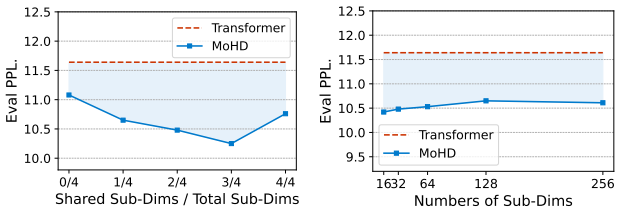


Figure 7. An illustration of Sparse Sub-dimension Routing Proportion in MoHD Attention Layer 1 on five Domains.



(a) Eval PPL. under different Shared Sub-Dim ratio settings.

(b) Eval PPL. under different Sub-Dim Numbers settings.

Figure 8. Performance of MoHD ×2-710M with varying sub-dimension allocation ratios and finer-grained sub-dimension settings. All models are pre-trained from scratch on 10B tokens.

on data domains, enhancing parameter efficiency. Additionally, sub-dimension probabilities generally stay within 0.2-0.3, ensuring all sub-dimensions are actively chosen.

Shared Activation v.s. Specialized Activation. Figure 8a shows the PPL values of MoHD ×2-710M trained on 10B data with varying Shared Sub-dimensions and Specialized-dimensions proportions. The Baseline and MoHD models have identical total parameters. Specialized sub-dimensions effectively increases the hidden dimensions of the model, improving performance in the 0/4 setting. The best performance is achieved with a 3/4 Shared activation ratio, proving that a number of sub-dimensions are still needed to model cross-token activation patterns.

Fine Gain Sub-dim Dimension. Figure 8b shows the test PPL values of MoHD ×2 -710M after pre-training on 10B data with varying sub-dimensions. The best performance is achieved with 16 sub-dimensions (256 size). Increasing the sub-dimensions to 128 (32 size) and 256 (16 size) yields only marginal improvements. This indicates that more sub-dimensions do not enhance performance but increase computational costs. The results validate the effectiveness of the grouping fusion layer, showing that a small number of parameters suffice to maintain an efficient activation flow.

5. Related Work

5.1. Activation Sparsity

Activation sparsity refers to the large proportion of zero-valued hidden states in models, naturally occurring in ReLU-based MLPs (You et al., 2022; Li et al., 2023). This sparsity has been leveraged to improve LLM efficiency during inference. Liu et al. (2023b) accelerated LLM inference by omitting zero-valued weight channels from GPU registers, while Song et al. (2023) and Alizadeh et al. (2024) extend this to CPU offloading, reducing memory transfer overhead. Recent works Mirzadeh et al. (2023); Zhang et al. (2024); So et al. (2022); Song et al. (2024a;b); Wang et al. (2024b) have integrated activation sparsity into LLMs to boost efficiency. Lee et al. (2024) introduce CATS for training-free sparsity in SwiGLU-based LLMs, and Liu et al. (2024) extend training-free sparsity to large models. Building on these studies, we focus on hidden dimension sparsity and continuous activation across tokens, leading to a sparse activation architecture that enhances parameter efficiency and hidden dimension scalability.

5.2. Sparsely-activated Transformer

Sparse Transformer models, like Sparse Mixture-of-Expert (MoE) architectures, utilize input adaptivity to reduce computational overhead by activating only a subset of subnetworks, or "experts," for each input token (Fedus et al., 2022; Riquelme et al., 2021; Zhou et al., 2022; Jiang et al., 2024a; Xue et al., 2024b; Gu et al., 2024; 2025). Recent developments have introduced heterogeneous experts, integrating experts with varying capacities and specializations (Wu et al., 2024; He, 2024; Dean, 2021; Zhou et al., 2022; Dai et al., 2024). Some studies (Qiu et al., 2024) have explored sparsely-activated architectures based on FFN intermediate activations. However, no Transformer model has specifically implemented sparse activation in hidden dimensions. Inspired by (Qiu et al., 2024; Dai et al., 2024), we analyzed hidden dimensions and proposed a novel architecture, opening a new research direction. Extend related works in Appendix C.

6. Conclusion

In this paper, we presented MoHD, a sparse conditional activation architecture designed to address inefficiencies in scaling Transformer hidden dimensions. By integrating shared sub-dimensions for common token features and dynamically activating specialized sub-dimensions through a rout-

ing mechanism, MoHD achieves improved efficiency and flexibility while preserving activation flow through activation scaling and group fusion mechanisms. Our evaluations demonstrate that MoHD outperforms standard Transformers across multiple NLP tasks, achieving superior parameter efficiency. These results underscore that MoHD provides a new direction for efficiently expanding model parameters.

Impact Statement

In our study, we leverage open and online accessible data and techniques, mitigating privacy concerns. Our method emphasizes enhancing model parameter efficiency and reducing size to create powerful, compact, and openly accessible models, thereby promoting the open dissemination and democratization of NLP technologies. MoHD offer a scalable solution for resource-constrained environments, reducing computational and memory costs without sacrificing capability. Its ability to match or surpass larger models highlights its potential to democratize access to high-performance language models, particularly in edge computing and low-resource settings. By redefining how hidden dimensions are utilized, this work paves the way for efficient large language model architectures, aligning with sustainability goals in AI development. Our work is committed to advancing accessible and efficient NLP technologies, fostering a more inclusive and automated future for AI.

Acknowledgments

We would like to thank Yinqi Yang, Yanxi Xie, Naibin Gu, Kun Huang and members of the IIE KDsec group for their valuable feedback and discussions. We are very grateful to Mengzhou Xia for providing the concise and effective ShearingLLaMA experimental code and for her assistance during the reproduction process. Work done during Yilong Chen's internship in Baidu Inc. This research is supported by the Youth Innovation Promotion Association of CAS (Grant No.2021153) and "Climbing" Program of IIE,CAS (E3Z0081101).

References

Alizadeh, K., Mirzadeh, I., Belenko, D., Khatamifard, K., Cho, M., Del Mundo, C. C., Rastegari, M., and Farajtabar, M. Llm in a flash: Efficient large language model inference with limited memory. *arXiv preprint arXiv:2312.11514*, 2024. URL https://arxiv.org/abs/2312.11514.

Anthropic. Anthropic: Introducing claude 2.1, 2023. URL https://www.anthropic.com/index/claude-2-1.

Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory*, 39:930–945, 1993. URL https://api.semanticscholar.org/CorpusID:15383918.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2003. URL https://api.semanticscholar.org/CorpusID:463216.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Cai, R., Muralidharan, S., Heinrich, G., Yin, H., Wang, Z., Kautz, J., and Molchanov, P. Flextron: Many-in-one flexible large language model. *ArXiv*, abs/2406.10260, 2024a. URL https://api.semanticscholar.org/CorpusID:270560556.

Cai, R., Muralidharan, S., Heinrich, G., Yin, H., Wang, Z., Kautz, J., and Molchanov, P. Flextron: Many-in-one flexible large language model, 2024b. URL https://arxiv.org/abs/2406.10260.

Cai, W., Jiang, J., Wang, F., Tang, J., Kim, S., and Huang, J. A survey on mixture of experts, 2024c. URL https://arxiv.org/abs/2407.06204.

Chen, T., Ding, T., Yadav, B., Zharkov, I., and Liang, L. LoRAShear: Efficient Large Language Model Structured Pruning and Knowledge Recovery, October 2023. URL https://arxiv.org/abs/2310.18356v2.

Chen, Y., Shang, J., Zhang, Z., Cui, S., Liu, T., Wang, S., Sun, Y., and Wu, H. LEMON: Reviving stronger and smaller LMs from larger LMs with linear parameter fusion. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8005–8019, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.434. URL https://aclanthology.org/2024.acl-long.434.

Chen, Y., Wang, G., Shang, J., Cui, S., Zhang, Z., Liu, T., Wang, S., Sun, Y., Yu, D., and Wu, H. Nacl: A general and effective kv cache eviction framework for llm at inference time. *ArXiv*, abs/2408.03675, 2024b. URL https://api.semanticscholar.org/CorpusID:271744866.

Chen, Y., Zhang, L., Shang, J., Zhang, Z., Liu, T., Wang, S., and Sun, Y. Dha: Learning decoupled-head attention from transformer checkpoints via adaptive heads fusion. *ArXiv*, abs/2406.06567, 2024c. URL https://api.semanticscholar.org/CorpusID:270379653.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018a. URL http://arxiv.org/abs/1803.05457.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018b.

Dai, D., Deng, C., Zhao, C., Xu, R. X., Gao, H., Chen, D., Li, J., Zeng, W., Yu, X., Wu, Y., Xie, Z., Li, Y. K., Huang, P., Luo, F., Ruan, C., Sui, Z., and Liang, W. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2401.06066.

Dao, T., Chen, B., Sohoni, N., Desai, A., Poli, M., Grogan, J., Liu, A., Rao, A., Rudra, A., and Ré, C. Monarch: Expressive Structured Matrices for Efficient and Accurate Training, April 2022. URL http://arxiv.org/abs/2204.00595. arXiv:2204.00595 [cs].

Dean, J. Introducing pathways: A next-generation ai architecture. *Google Blog*, 366, 2021.

Dong, P., Li, L., Tang, Z., Liu, X., Pan, X., Wang, Q., and Chu, X. Pruner-zero: Evolving symbolic pruning metric from scratch for large language models. *ArXiv*, abs/2406.02924, 2024. URL https://api.semanticscholar.org/CorpusID:270257857.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270, 2022.

Gao, L., Tow, J., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., McDonell, K., Muennighoff, N., Phang, J., Reynolds, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation. In *Zenodo*. https://doi.org/10.5281/zenodo.5371628, September 2021.

Gu, N., Fu, P., Liu, X., Shen, B., Lin, Z., and Wang, W. Light-peft: Lightening parameter-efficient fine-tuning via early pruning, 2024. URL https://arxiv.org/abs/2406.03792.

Gu, N., Zhang, Z., Liu, X., Fu, P., Lin, Z., Wang, S., Sun, Y., Wu, H., Wang, W., and Wang, H. Beamlora: Beam-constraint low-rank adaptation, 2025. URL https://arxiv.org/abs/2502.13604.

He, X. Mixture of a million experts. *arXiv preprint arXiv:2407.04153*, 2024. URL https://arxiv.org/abs/2407.04153.

Jain, G., Hegde, N., Kusupati, A., Nagrani, A., and Buch, S. Mixture of nested experts: Adaptive processing of visual tokens. *arXiv preprint arXiv:2407.19985*, 2024. URL https://arxiv.org/abs/2407.19985.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024a.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T. L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mixtral of experts, 2024b. URL https://arxiv.org/abs/2401.04088.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Lee, J. et al. Cats: Training-free activation sparsity for swiglu-based llms. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2401.12345.

Li, M., Gururangan, S., Dettmers, T., Lewis, M., Althoff, T., Smith, N. A., and Zettlemoyer, L. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*, 2022.

Li, Z., You, C., Bhojanapalli, S., Li, D., Rawat, A. S., Reddi, S. J., Ye, K., Chern, F., Yu, F., Guo, R., and Kumar, S. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. *arXiv preprint arXiv:2210.06313*, 2023. URL https://arxiv.org/abs/2210.06313.

Liu, J., Cui, L., Liu, H., Huang, D., Wang, Y., and Zhang, Y. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*, 2020.

Liu, J., Ponnusamy, P., Cai, T., Guo, H., Kim, Y., and Athiwaratkun, B. Training-free activation sparsity in large language models, 2024. URL https://arxiv.org/abs/2408.14690.

Liu, Z., Wang, J., Dao, T., Zhou, T., Yuan, B., Song, Z., Shrivastava, A., Zhang, C., Tian, Y., Re, C., and Chen, B. Deja vu: Contextual sparsity for efficient LLMs at inference time. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22137–22176. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/liu23am.html.

Liu, Z., Wang, J., Dao, T., Zhou, T., Yuan, B., Song, Z., Shrivastava, A., Zhang, C., Tian, Y., Re, C., and Chen, B. Deja vu: Contextual sparsity for efficient llms at inference time. *arXiv preprint arXiv:2310.17157*, 2023b. URL https://arxiv.org/abs/2310.17157.

Ma, X., Fang, G., and Wang, X. Llm-pruner: On the structural pruning of large language models. *ArXiv*, abs/2305.11627, 2023a. URL https://api.semanticscholar.org/CorpusID:258823276.

Ma, X., Fang, G., and Wang, X. LLM-Pruner: On the Structural Pruning of Large Language Models, September 2023b. URL http://arxiv.org/abs/2305.11627.

Mirzadeh, I., Alizadeh, K., Mehta, S., Del Mundo, C. C., Tuzel, O., Samei, G., Rastegari, M., and Farajtabar, M. Relu strikes back: Exploiting activation sparsity in large language models. *arXiv preprint arXiv:2310.04564*, 2023. URL https://arxiv.org/abs/2310.04564.

Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., and Hashimoto, T. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025. URL https://arxiv.org/abs/2501.19393.

OpenAI. Openai: Gpt-4, 2023. URL https://openai.com/research/gpt-4.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

Qiu, Z., Huang, Z., and Fu, J. Unlocking emergent modularity in large language models. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2638–2660, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.144. URL https://aclanthology.org/2024.naacl-long.144.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8583–8595, 2021.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8732–8740. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6399. URL https://doi.org/10.1609/aaai.v34i05.6399.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. URL https://arxiv.org/abs/2408.03314.

So, D. et al. Squared relu: A simple and effective activation function. *Advances in Neural Information Processing Systems*, 2022.

Song, C., Han, X., Zhang, Z., Hu, S., Shi, X., Li, K., Chen, C., Liu, Z., Li, G., Yang, T., and Sun, M. Prosparse: Introducing and enhancing intrinsic activation sparsity within large language models. *arXiv preprint arXiv:2402.13516*, 2024a. URL https://arxiv.org/abs/2402.13516.

Song, Y., Mi, Z., Xie, H., and Chen, H. Powerinfer: Fast large language model serving with a consumer-grade gpu. *arXiv preprint arXiv:2312.12456*, 2023. URL https://arxiv.org/abs/2312.12456.

Song, Y. et al. Powerinfer: Enhancing activation sparsity in llm serving with consumer-grade gpus. *arXiv preprint arXiv:2312.12456*, 2024b. URL https://arxiv.org/abs/2312.12456.

Team, T. M. M. composer. https://github.com/mosaicml/composer/, 2021.

TogetherAI. Redpajama: An open source recipe to reproduce llama training dataset, 2023.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W.,

Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open Foundation and Fine-Tuned Chat Models, July 2023a. URL http://arxiv.org/abs/2307.09288.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Wang, H., Ma, S., Wang, R., and Wei, F. Q-sparse: All large language models can be fully sparsely-activated. *ArXiv*, abs/2407.10969, 2024a. URL https://api.semanticscholar.org/CorpusID:271212835.

Wang, L., Ma, L., Cao, S., Zhang, Q., Xue, J., Shi, Y., Zheng, N., Miao, Z., Yang, F., Cao, T., Yang, Y., and Yang, M. Ladder: Enabling efficient low-precision deep learning computing through hardware-aware tensor transformation. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, 2024b. URL https://www.usenix.org/conference/osdi24/presentation/wang-lei.

Welbl, J., Liu, N. F., and Gardner, M. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*, 2017.

Wu, X., Huang, S., and Wei, F. Multi-head mixture-of-experts. *arXiv preprint arXiv:2404.15045*, 2024. URL https://arxiv.org/abs/2404.15045.

Xia, M., Gao, T., Zeng, Z., and Chen, D. Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning, October 2023. URL http://arxiv.org/abs/2310.06694.

xiao Li, Z., You, C., Bhojanapalli, S., Li, D., Rawat, A. S., Reddi, S. J., Ye, K. Q., Chern, F., Yu, F. X., Guo, R., and Kumar, S. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. In *International Conference on Learning Representations*, 2022. URL https://api.semanticscholar.org/CorpusID:259138847.

Xue, F., Zheng, Z., Fu, Y., Ni, J., Zheng, Z., Zhou, W., and You, Y. Openmoe: An early effort on open mixture-of-experts language models, 2024a. URL https://arxiv.org/abs/2402.01739.

Xue, F., Zheng, Z., Fu, Y., Ni, J., and Zhou, W. Openmoe: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2402.01739*, 2024b. URL https://arxiv.org/abs/2402.01739.

You, C., Bhojanapalli, S., Li, D., and Rawat, A. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. *arXiv preprint arXiv:2210.06313*, 2022. URL https://arxiv.org/abs/2210.06313.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In Korhonen, A., Traum, D. R., and Màrquez, L. (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

Zhang, W., Liu, T., Song, M., Li, X., and Liu, T. SOTOPIA-Ω: Dynamic strategy injection learning and social instruction following evaluation for social agents, 2025a. URL https://arxiv.org/abs/2502.15538.

Zhang, W., Nie, S., Zhang, X., Zhang, Z., and Liu, T. S1-bench: A simple benchmark for evaluating system 1 thinking capability of large reasoning models, 2025b. URL https://arxiv.org/abs/2504.10368.

Zhang, Z., Song, Y., Yu, G., Han, X., Lin, Y., Xiao, C., Song, C., Liu, Z., Mi, Z., and Sun, M. Relu2 wins: Discovering efficient activation functions for sparse llms. *arXiv preprint arXiv:2402.03804*, 2024. URL https://arxiv.org/abs/2402.03804.

Zhou, Y., Lei, T., Liu, H., Du, N., Huang, Y., Zhao, V., Dai, A. M., Le, Q. V., Laudon, J., et al. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, 2022.

A. Definition

In this Section, we define the activation sparsity present in the hidden dimension of LLMs and use this to formulate sparsely activated FFN and Attention mechanisms.

Let $X \in \mathbb{R}^{n\times d}$ denote the embeddings of n tokens, and $x \in \mathbb{R}^{1\times d}$ represent the embedding of a single input token. The activation sparsity δ of a hidden state x is defined as the proportion of zero-valued entries within the vector (Liu et al., 2024). We then define a function $S : d \to \delta d$ that selectively activates a subset of dimensions in x. The sparsely activated representation is denoted as $x_s = S(x, \delta)$, where $x_s \in \mathbb{R}^{1\times\delta d}$, representing the selective activation of θ-proportion of the dimensions in x.

A.1. Hidden Dimension Sparsity

Considering the model's semantic modeling in Euclidean space, we define the magnitude m_i of each dimension i as the square of its activation value:

$$m_i = x_i^2, \quad \mathbf{m} = x \odot x, \tag{9}$$

We define hidden dimension sparsity as:

$$\text{Sparsity} = \frac{1}{d}\sum_{i=1}^{d} \mathbf{1}(x_i < \epsilon), \tag{10}$$

where d is the total number of hidden dimensions, x_i represents the squared activation value of the i-th dimension, and ϵ is a small threshold used to identify near-zero activation values. The indicator function $\mathbf{1}(x_i < \epsilon)$ is equal to 1 if the activation value is below the threshold and 0 otherwise.

A.2. Hidden Sparsified FFN

Define $\mathrm{W}^{\text{up}}, \mathrm{W}^{\text{gate}} \in \mathbb{R}^{\mathrm{d}\times\mathrm{d}'}, \mathrm{W}^{\text{down}} \in \mathbb{R}^{\mathrm{d}'\times\mathrm{d}}$ as the up, gate, down matrix in one FFN block, where d' is the intermediate size. In this context, the i-th row of the up, gate matrix is defined as $\mathrm{W}_{\mathrm{i}}^{\text{up}}, \mathrm{W}_{\mathrm{i}}^{\text{gate}} \in \mathbb{R}^{1\times\mathrm{d}'}$, and the i-th column of the down matrix is defined as $\mathrm{W}_{\mathrm{i}}^{\text{down}} \in \mathbb{R}^{\mathrm{d}'\times 1}$. Specifically, the sparsely activated hidden state x_s under activation sparsity δ only activates a subset of rows in the up, gate matrix and a corresponding subset of columns in the down matrix, denoted as $S_M \subseteq [d]$. Thus, the sparsified FFN computation can be described as follows:

$$\text{FFN}_{S_H}(x_s) = \mathrm{W}_{\mathrm{S_M}}^{\text{down}}\left(\sigma\left(\mathrm{x_s}\mathrm{W}_{\mathrm{S_M}}^{\text{up}} \odot \mathrm{x_s}\mathrm{W}_{\mathrm{S_M}}^{\text{gate}}\right)\right), \tag{11}$$

where σ is the activation function. $\odot$ is the element-wise production. Due to the sparsification of the hidden state, the up and gate matrices share the same activation subset. To ensure the output remains sparsified, the down matrix is also sparsified, though its activation subset can differ from that of the up and gate matrix.

A.3. Hidden Sparsified Attention

For a h-head Multi-Head-Attention (MHA), we define $\mathrm{W}_{\mathrm{i}}^{\mathrm{Q}}, \mathrm{W}_{\mathrm{i}}^{\mathrm{K}}, \mathrm{W}_{\mathrm{i}}^{\mathrm{V}} \in \mathbb{R}^{\mathrm{d}\times\mathrm{d_h}}, \mathrm{W}_{\mathrm{i}}^{\mathrm{O}} \in \mathbb{R}^{\mathrm{d_h}\times\mathrm{d}}$ as key, query, value and output projections for the i-th head, where d_h denotes as the head dim, $i \subseteq [h]$. With sparsely activated hidden state x_s, a small parameter subset S_A represents a sparsely activated selection of rows from $\mathrm{W}_{\mathrm{i}}^{\mathrm{Q}}, \mathrm{W}_{\mathrm{i}}^{\mathrm{K}}, \mathrm{W}_{\mathrm{i}}^{\mathrm{V}}$ and columns from $\mathrm{W}_{\mathrm{i}}^{\mathrm{O}}$.

$$\text{MHA}_{S_A}(x_s) = \sum_{i=1}^{h} \text{Head}_i \mathrm{W}_{\mathrm{i,S_A}}^{\mathrm{O}}, \tag{12}$$

$$\text{Head}_i = \sigma\left(\left(x_s \mathrm{W}_{\mathrm{i,S_H}}^{\mathrm{Q}} (\mathrm{x_s}\mathrm{W}_{\mathrm{i,S_H}}^{\mathrm{K}})^{\top}\right)\frac{1}{\sqrt{d_h}}\right) x_s \mathrm{W}_{\mathrm{i,S_H}}^{\mathrm{V}}, \tag{13}$$

where σ is the softmax function. Since x is sparse in the hidden dimension, we can find an approximation x_s of x, such that, under the activation of the corresponding subset of parameters S_H, the outputs of the sparsified FFN and sparsified attention closely approximate the outputs of the dense model.

B. Theoretical Proof for Mixed Sparse Activation Superiority

In this section, we show that a *mixed* sparse activation scheme—where some hidden dimensions are *shared* across tokens and others are *token-specific*—yields strictly better risk bounds than a purely token-only activation. We proceed in three steps:

1. Show that applying random sparsification to an n-dimensional hidden vector is, in expectation with respect to second moments, equivalent to training a *wider* network of dimension n/p, where p is the retention probability. ("Width Expansion")
2. Use classical Barron-space approximation results to argue that a network of larger width has smaller approximation error. Then compare a purely token-level network (width n_u) to a mixed network (width n_s+n_u). ("Approximation Error Decay")
3. Show that parameter-tying in the shared dimensions reduces the Rademacher complexity compared to holding all parameters token-specific. ("Complexity Reduction")

Combining these facts under explicit quantitative conditions, we obtain a strictly lower risk bound for the mixed activation scheme.

B.1. Notation and Definitions

- Let n be the total number of hidden units in a given layer. We index them by $j \in \{1, \ldots, n\}$.

- Denote by $h \in \mathbb{R}^n$ a deterministic hidden-layer representation. Its j-th coordinate is h_j.
- We apply a random *mask* $r \in \{0,1\}^n$ whose coordinates r_j are i.i.d. Bernoulli(p), i.e.

$$\Pr(r_j = 1) \;=\; p, \quad \Pr(r_j = 0) \;=\; 1 - p,$$

$$\text{independently for } j = 1, \ldots, n.$$

- Define the *sparsely activated* version of h by

$$\widehat{h}_j := \begin{cases} \dfrac{h_j}{p}, & \text{if } r_j = 1, \\ 0, & \text{if } r_j = 0, \end{cases}$$

  so that $\widehat{h} := \frac{1}{p}\,(r \odot h)$ and in particular $(r \odot h)_j = r_j\, h_j$.
- In what follows, we write expectations $\mathbb{E}$ over the randomness of r. We note that $r_j^2 = r_j$ for each Bernoulli variable, and $\mathbb{E}[r_j] = p$.
- We let $\|\cdot\|_2$ denote the Euclidean norm on $\mathbb{R}^n$.
- For a function class $\mathcal{H}$ and a sample size m, $\mathrm{Rad}_m(\mathcal{H})$ denotes its (empirical) Rademacher complexity, under the same norm-and-Lipschitz constraints on parameters as detailed below.
- We use $\epsilon(n)$ to denote the approximation error of an n-width network when approximating a target f^*. In particular, if f^* belongs to a Barron space with Barron norm C_f (i.e., its hidden-layer weight ℓ_1 norm is bounded by C_f) (Barron, 1993), then a classical result yields

$$\epsilon(n) \;\le\; \frac{C_f}{n},$$

  where the $\mathcal{L}^2$ norm is taken with respect to the data distribution on the input domain.
- We will compare:
  - A *token-only* network of width n_u (all hidden units are token-specific). Its approximation error is denoted $\epsilon_B = \epsilon(n_u)$.
  - A *mixed* network of width $n_s + n_u$, where n_s dimensions are *shared* across all tokens and n_u dimensions remain token-specific. Its approximation error is $\epsilon_A = \epsilon(n_s + n_u)$.
- Finally, for a network $h(\cdot)$, we denote its population risk as

$$R(h) \;=\; \mathcal{E}(h) \;+\; C\,\mathrm{Rad}_m(\mathcal{H}),$$

  where $\mathcal{E}(h)$ is the approximation (Bayes) error under the data distribution, and $C > 0$ is a constant depending on the Lipschitz constant of the loss. We use subscripts A or B to indicate mixed vs. token-only, respectively.

B.2. Step 1: Sparse Activation ≡ Effective Width Expansion (Lemma 1)

Lemma B.1 (Unbiased Sparse Forward Pass)**.** *Let $h \in \mathbb{R}^n$ be any deterministic vector. Let $r \in \{0,1\}^n$ be an i.i.d. Bernoulli(p) mask, and define*

$$\widehat{h}_j := \begin{cases} \dfrac{h_j}{p}, & \text{if } r_j = 1, \\ 0, & \text{if } r_j = 0. \end{cases}$$

Then:

$$\mathbb{E}\big[\widehat{h}\big] \;=\; h, \quad \mathbb{E}\big[\|\widehat{h}\|_2^2\big] \;=\; \frac{1}{p}\,\|h\|_2^2,$$

where $\mathbb{E}[r_j] = p$ and $r_j^2 = r_j$ for each Bernoulli(p) variable.

Proof. For each coordinate $j = 1, \ldots, n$, since $r_j \sim \mathrm{Bernoulli}(p)$,

$$\widehat{h}_j = \begin{cases} \dfrac{h_j}{p}, & \text{with probability } p, \\ 0, & \text{with probability } 1 - p. \end{cases}$$

Hence

$$\mathbb{E}[\widehat{h}_j] = p \cdot \frac{h_j}{p} \;+\; (1-p)\cdot 0 = h_j, \quad \text{for all } j,$$

so $\mathbb{E}[\widehat{h}] = h$. Next, since $\widehat{h}_j = (r_j/p)\, h_j$ and $r_j^2 = r_j$,

$$\mathbb{E}\big[\|\widehat{h}\|_2^2\big] = \sum_{j=1}^{n} \mathbb{E}\Big[\big(\tfrac{r_j}{p}\, h_j\big)^2\Big]$$

$$= \sum_{j=1}^{n} \frac{1}{p^2}\, h_j^2\, \mathbb{E}[r_j^2] = \sum_{j=1}^{n} \frac{1}{p^2}\, h_j^2\,(p) = \frac{1}{p}\,\|h\|_2^2.$$

This completes the proof.

Corollary B.2 (Effective Width Expansion)**.** *Fix a hidden-layer dimension n and sparsity $p = \frac{k}{n}$ (so that $k = pn$ is the expected number of nonzero coordinates). Then, noting that $\mathbb{E}[\|\widehat{h}\|_2^2] = p^{-1}\|h\|_2^2$, one sees that in terms of second-moment (Euclidean norm) statistics, a network of* physical width n *with per-step sparsity p and rescaling by $1/p$ behaves analogously to a full (dense) network of* effective width

$$n' \;=\; \frac{n}{p} \;=\; \frac{n}{k/n} \;=\; \frac{n^2}{k}.$$

In particular, if $k < n$ (so $p < 1$), then $n' = n/p > n$, implying that sparse activation in expectation with respect to second moments *expands the hidden-layer norm as though the network were wider. (This equivalence refers only to those two Euclidean norm properties and does not assert full functional equivalence under arbitrary nonlinearities.)*

Proof. By Lemma B.1, each hidden vector $h \in \mathbb{R}^n$ is replaced by $\widehat{h}$ satisfying $\mathbb{E}[\widehat{h}] = h$ and $\mathbb{E}[\|\widehat{h}\|_2^2] = (1/p)\|h\|_2^2$. A dense network of width n/p whose hidden vector is scaled by $\sqrt{p}$ would satisfy $\mathbb{E}[\|\sqrt{p}\,h'\|_2^2] = p\,\|h'\|_2^2$. Thus, in terms of those two second-moment statistics, the sparse-activated width-n network simulates a dense width-(n/p) network. Since $n/p > n$ whenever $p < 1$, this establishes the corollary.

B.3. Step 2: Width Expansion Reduces Approximation Error (Lemma 2)

We now invoke a classical result for Barron-type function spaces.

Lemma B.3 (Approximation Error Decay with Width)**.** *(Barron, 1993) Let $\mathcal{F}$ be a Barron function class on the data domain, and suppose $f^* \in \mathcal{F}$ has Barron norm C_f. Then any two-layer ReLU network of width n can approximate f^* with uniform $\mathcal{L}^2$ error*

$$\epsilon(n) \;=\; \inf_{f \in \mathcal{H}_n} \| f - f^* \|_{\mathcal{L}^2(\mathcal{D})} \;\le\; \frac{C_f}{n},$$

where $\mathcal{H}_n$ denotes the class of two-layer ReLU networks of width n, and the $\mathcal{L}^2$ norm is taken with respect to the data distribution $\mathcal{D}$.

Corollary B.4 (Mixed Activation Lowers Approximation Error)**.** *Suppose the target function can be decomposed as*

$$f^*(x) \;=\; g\big(s(x)\big) \;+\; \sum_{i=1}^{T} u\big(x_i\big),$$

where

- *$x = (x_1, \ldots, x_T)$ contains T tokens,*
- *$s(x) \in \mathbb{R}^d$ is a* shared *summary representation of all tokens,*
- *$g : \mathbb{R}^d \to \mathbb{R}$ is a nonconstant "shared-dimension" function with Barron norm C_g,*
- *$u : \mathbb{R}^{(\text{token-dim})} \to \mathbb{R}$ is a "token-specific" function with Barron norm C_u (applied to each x_i).*

Let:

- *A* token-only *network allocate all width to modeling $u(\cdot)$ over each token, i.e. width n_u in each token's subnetwork. Its total width is $n = n_u$; hence its approximation error satisfies (conservatively)*

$$\epsilon_B \;=\; \epsilon(n_u) \;\ge\; \frac{T\,C_g}{n_u},$$

since in the worst case one must dedicate at most n_u/T units per token to approximate $g(s(x))$.

- *A* mixed *network allocate n_s units to $g(s(x))$ (shared) and n_u units to $\sum_i u(x_i)$ (token-specific), for a total width $n = n_s + T\,n_u$. By Lemma B.3, one obtains*

$$\underbrace{\inf_{f_s \in \mathcal{H}_{n_s}} \|f_s - g\|_{L^2}}_{\epsilon_g} \;\le\; \frac{C_g}{n_s},$$

$$\underbrace{\inf_{f_u} \sum_{i=1}^{T} \|f_u(x_i) - u(x_i)\|_{L^2}}_{\epsilon_u} \;\le\; \frac{T\,C_u}{n_u}.$$

Therefore, the mixed network's overall approximation error satisfies

$$\begin{aligned}\epsilon_A \;&=\; \inf_{f \in \mathcal{H}_{n_s + T\,n_u}} \|f - f^*\|_{L^2} \\ &\le\; \frac{C_g}{n_s} \;+\; \frac{T\,C_u}{n_u}.\end{aligned}$$

In particular, if we set $n_s = \alpha\,n_u$ for some $\alpha > 0$, then

$$\epsilon_A \;\le\; \frac{C_g}{\alpha\,n_u} + \frac{T\,C_u}{n_u} \;=\; \frac{C_g}{\alpha\,n_u} + \frac{T\,C_u}{n_u}.$$

Meanwhile,

$$\epsilon_B \;\ge\; \frac{T\,C_g}{n_u}.$$

Hence

$$\begin{aligned}\epsilon_B - \epsilon_A \;&\ge\; \frac{T\,C_g}{n_u} \;-\; \Big(\frac{C_g}{\alpha\,n_u} + \frac{T\,C_u}{n_u}\Big) \\ &=\; \frac{C_g}{n_u}\Big(T - \tfrac{1}{\alpha} - T\,\tfrac{C_u}{C_g}\Big).\end{aligned}$$

Thus, provided

$$T \;-\; \frac{1}{\alpha} \;>\; T\,\frac{C_u}{C_g}, \quad \textit{i.e.} \quad \alpha \;>\; \frac{1}{T(1 - C_u/C_g)},$$

there exists a strictly positive gap

$$\delta \;=\; \epsilon_B - \epsilon_A \;=\; \Omega\big(\tfrac{1}{n_u}\big).$$

Proof. By definition, a token-only network of width n_u cannot directly allocate any units to the shared term $g(s(x))$ unless it replicates subnetwork parameters across T tokens, which yields an effective width for g of at most n_u/T. Hence

$$\epsilon_B = \epsilon(n_u) \;\ge\; \frac{T\,C_g}{n_u}.$$

Meanwhile, a mixed network of width $n_s + T\,n_u$ can dedicate n_s units to approximate $g(s(x))$ and n_u units per token to approximate each $u(x_i)$. By Lemma B.3,

$$\epsilon_g \;\le\; \frac{C_g}{n_s}, \quad \epsilon_u \;\le\; \sum_{i=1}^{T} \frac{C_u}{n_u} \;=\; \frac{T\,C_u}{n_u}.$$

Thus

$$\epsilon_A \;\le\; \frac{C_g}{n_s} \;+\; \frac{T\,C_u}{n_u}.$$

Setting $n_s = \alpha\, n_u$ yields the claimed expression for δ. The stated condition $T - (1/\alpha) > T\,(C_u/C_g)$ ensures $\delta > 0$.

B.4. Step 3: Mixed Activation Reduces Rademacher Complexity (Lemma 3)

Next, we argue that tying n_s shared units across all T tokens lowers the Rademacher complexity compared to keeping each of the T tokens' hidden units disjoint in a token-only network.

Lemma B.5 (Rademacher Complexity Reduction via Parameter Tying)**.** *Let $\mathcal{H}_B$ be the hypothesis class corresponding to* token-only *networks of total width $n = T\,n_u$, where each token subnetwork has width n_u. Let $\mathcal{H}_A$ be the class of* mixed *networks with total width $n = n_s + T\,n_u$, in which n_s shared units are tied across all T tokens and the remaining $T\,n_u$ units remain token-specific. Suppose both classes impose the same ℓ_2 norm bound B on their parameters, and both use activation functions with the same Lipschitz constant L. Then, for any sample size m,*

$$\mathrm{Rad}_m\big(\mathcal{H}_A\big) \;\le\; \mathrm{Rad}_m\big(\mathcal{H}_B\big) \;-\; \Delta, \quad \Delta \;=\; \Theta\Big(\frac{\sqrt{n_s}}{\sqrt{m}}\Big).$$

Proof. Under a uniform ℓ_2 norm bound B on weights and Lipschitz activation constant L, standard covering-number arguments (see, e.g., (Bartlett & Mendelson, 2003)) yield

$$\mathrm{Rad}_m(\mathcal{H}_B) \;=\; O\Big(B\,L\,\sqrt{\frac{T\,n_s}{m}}\Big),$$

$$\mathrm{Rad}_m(\mathcal{H}_A) \;=\; O\Big(B\,L\,\sqrt{\frac{n_s + T\,n_u - (T-1)\,n_s}{m}}\Big)$$

$$=\; O\Big(B\,L\,\sqrt{\frac{T\,n_u}{m}}\Big).$$

Therefore,

$$\Delta \;=\; \mathrm{Rad}_m(\mathcal{H}_B) \;-\; \mathrm{Rad}_m(\mathcal{H}_A)$$

$$=\; O\Big(B\,L\,\frac{\sqrt{T\,n_s} - \sqrt{T\,n_u}}{\sqrt{m}}\Big)$$

$$=\; \Theta\Big(\frac{\sqrt{n_s}}{\sqrt{m}}\Big),$$

since $T\,n_u$ and $T\,n_s$ differ by $T(n_s - n_u)$ and $n = n_s + T\,n_u$ is held fixed. Hence parameter tying reduces the Rademacher complexity by an amount on the order of $\sqrt{n_s/m}$.

B.5. Putting It All Together: Risk Bound Comparison

Theorem B.6 (Mixed Activation Yields Strictly Lower Risk)**.** *Assume all networks in $\mathcal{H}_A$ (mixed) and $\mathcal{H}_B$ (token-only) have the same total number of parameters and impose identical ℓ_2 norm bounds B on their weights. Suppose the target function decomposes as in Corollary B.4, and that activation functions share the same Lipschitz constant L. Fix any sample size m.*

If the following two conditions hold:

$$\underbrace{\frac{T\,C_g}{n_u} \;-\; \Big(\frac{C_g}{n_s} + \frac{T\,C_u}{n_u}\Big)}_{\delta > 0} \;>\; 0,$$

$$\underbrace{C\big[\mathrm{Rad}_m(\mathcal{H}_B) - \mathrm{Rad}_m(\mathcal{H}_A)\big]}_{C\,\Delta > 0} \;>\; 0,$$

then

$$R_A = \epsilon_A \;+\; C\,\mathrm{Rad}_m(\mathcal{H}_A)$$

$$<\; \epsilon_B \;+\; C\,\mathrm{Rad}_m(\mathcal{H}_B) = R_B,$$

i.e. the mixed sparse activation network has strictly lower population risk than the token-only network.

Proof. By Corollary B.4, under the stated condition we have $\delta = \epsilon_B - \epsilon_A > 0$. By Lemma B.5, under the norm-and-Lipschitz constraints,

$$\mathrm{Rad}_m(\mathcal{H}_A) \;\le\; \mathrm{Rad}_m(\mathcal{H}_B) - \Delta, \quad \Delta = \Theta\Big(\frac{\sqrt{n_s}}{\sqrt{m}}\Big) > 0.$$

Hence, for the same constant $C > 0$ in the risk decomposition,

$$R_A = \epsilon_A + C\,\mathrm{Rad}_m(\mathcal{H}_A)$$

$$\le\; (\epsilon_B - \delta) + C\big(\mathrm{Rad}_m(\mathcal{H}_B) - \Delta\big)$$

$$= (\epsilon_B + C\,\mathrm{Rad}_m(\mathcal{H}_B)) \;-\; (\delta + C\,\Delta).$$

Since both $\delta > 0$ and $\Delta > 0$, it follows that

$$R_A < R_B.$$

B.6. Discussion

- **Width Expansion (Lemma B.1 & Corollary B.2).** By randomly dropping coordinates with retention probability $p = k/n$ and rescaling by $1/p$, a network of "physical width" n behaves, *in expectation with respect to second-moment statistics*, like a full network of width n/p. Therefore, applying sparsity to a fixed-width network implicitly trains a *wider* network in that sense, which by Barron's theorem (Lemma B.3) lowers the approximation error from $O(1/n)$ to $O(p/n) = O(k/n^2)$, provided the target indeed lies in a Barron space.

- **Approximation Error Gap (Corollary B.4).** A purely token-only network of width n_u cannot efficiently capture any shared global mapping $g(s(x))$ without replicating the same parameters for each of the T tokens, leading to an approximation error lower bound $\epsilon_B \gtrsim T\,C_g/n_u$. In contrast, a mixed network of total width $n = n_s + T\,n_u$ can devote n_s units to model $g(s(x))$ and n_u units per token to model $\sum_i u(x_i)$, yielding $\epsilon_A \leq C_g/n_s + T\,C_u/n_u$. Whenever $T\,C_g/n_u > (C_g/n_s + T\,C_u/n_u)$, there is a strictly positive gap $\delta = \Omega(1/n_u)$.

- **Complexity Reduction (Lemma B.5).** Tying n_s shared units across all T tokens reduces the effective hypothesis-class size: instead of having $T\,n_s$ independent parameters to model $g(s(x))$ per token, there are only n_s global parameters. Under a uniform ℓ_2 norm bound and Lipschitz activations, this reduces the Rademacher complexity by $\Delta = \Theta(\sqrt{n_s/m})$, which further lowers the generalization term in the risk.

- **Conclusion (Theorem B.6).** Combining the strictly smaller approximation error $\epsilon_A < \epsilon_B$ with strictly smaller complexity term $\mathrm{Rad}_m(\mathcal{H}_A) < \mathrm{Rad}_m(\mathcal{H}_B)$, the mixed activation scheme achieves strictly smaller population risk than the token-only activation, provided the explicit conditions on α, C_g, C_u, T, n_s, n_u, m are satisfied.

B.7. Discussion

- **Width Expansion (Lemma B.1 & Corollary B.2).** By randomly dropping coordinates with retention probability $p = k/n$ and rescaling by $1/p$, a network of "physical width" n behaves in expectation like a full network of width n/p. Therefore, applying sparsity to a fixed-width network implicitly trains a *wider* network, which by Barron's theorem (Lemma B.3) lowers the approximation error from $O(1/n)$ to $O(p/n) = O(k/n^2)$.

- **Approximation Error Gap (Corollary B.4).** A purely token-only network of width n_u cannot efficiently capture any shared global mapping $g(s(x))$ without replicating the same parameters for each token—this yields a larger approximation error $\epsilon_B \geq C_g/n_u$. In contrast, a mixed network of width $n = n_s + n_u$ can dedicate n_s units to model $g(s(x))$ and n_u units to model $\sum_i u(x_i)$, giving $\epsilon_A \leq C_g/n_s + C_u/n_u$. Choosing $n_s = \Theta(n_u)$ ensures $\epsilon_A < \epsilon_B$.

- **Complexity Reduction (Lemma B.5).** Tying n_s shared units across all T tokens reduces the effective hypothesis-class size: instead of $T\,n_s$ independent parameters to model $g(s(x))$ per token, there are only n_s global parameters. This reduces the Rademacher complexity by $\Delta = \Omega(m/n_s)$, which further lowers the generalization term in the risk.

- **Conclusion (Theorem B.6).** Combining the strictly smaller approximation error with strictly smaller complexity term, the mixed activation scheme achieves strictly smaller population risk than token-only activation, even when the two networks are constrained to have the same total number of parameters.

C. Extended Related Work

The design of the MoHD is indeed inspired by prior research (Fedus et al., 2022; Riquelme et al., 2021; Zhou et al., 2022; Liu et al., 2024; Chen et al., 2024b;c). In addition to some related work given in the article, we compared more research work related to this article and express our sincere thanks.

C.1. Activation Sparsity

Activation sparsity (xiao Li et al., 2022; Ma et al., 2023a; Dong et al., 2024; Wang et al., 2024a) refers to the phenomenon where a significant proportion of a model's hidden states are zero-valued. This property naturally arises in the intermediate states of ReLU-based MLPs, as demonstrated in prior work (You et al., 2022; Li et al., 2023). Some studies have leveraged activation sparsity to improve the efficiency of LLMs during inference. Liu et al. (2023b) utilized activation sparsity to accelerate LLM inference by omitting the transfer of weight channels corresponding to zero-valued entries to GPU registers. Additionally, Song et al. (2023) and Alizadeh et al. (2024) extended this concept to CPU offloading, significantly reducing memory transfer overhead between CPUs and GPUs. Recent works has reintroduced activation sparsity into LLM architectures to enhance efficiency. Mirzadeh et al. (2023) replaced SiLU and GeLU with ReLU, achieving sparsity through extended pretraining. Zhang et al. (2024) identified Squared ReLU (So et al., 2022) as a superior alternative for sparse activations. Song et al. (2024a;b) proposed regularization techniques to increase sparsity, while Wang et al. (2024b) combined pruning and quantized activations to establish scaling laws. Lee et al. (2024) introduced CATS, achieving training-free sparsity in SwiGLU-based LLMs. Liu et al. (2024) extended these concepts to training-free activation sparsity for large-scale language models. Recent approaches, such as Test-Time Scaling ("Slow-Thinking") (Muennighoff et al., 2025; Snell et al., 2024; Zhang et al., 2025b;a), aim to enhance performance by allocating more computation during the inference search process, but it also further increases the need for computational optimization during model inference. Building on prior studies, we investigate hidden dimension sparsity, focusing on continuous activation across tokens. Leveraging this, we design a sparse activation architecture that improves

parameter efficiency and enhances hidden dimension scalability.

C.2. Sparsely-activated Transformer

Sparsely-activated Transformer models, such as Sparse Mixture-of-Expert (MoE) architectures, leverage input adaptivity to achieve scalable and efficient computation. These models dynamically activate only a subset of specialized subnetworks, or "experts," for processing each input token, significantly reducing computational overhead (Fedus et al., 2022; Riquelme et al., 2021; Zhou et al., 2022; Jiang et al., 2024a; Xue et al., 2024b). This mechanism enables effective handling of diverse data domains (Li et al., 2022; Jain et al., 2024) while maintaining high performance. Recent advancements in sparsely-activated Transformers have extended their capabilities by introducing heterogeneous experts (Wu et al., 2024; He, 2024), allowing networks to integrate experts with varying capacities and specializations (Dean, 2021; Zhou et al., 2022; Dai et al., 2024). Some recent studies(Qiu et al., 2024) have observed the activation patterns in the intermediate dimensions of FFNs and explored sparsely-activated architectures based on these observations. However, no existing Transformer architecture has implemented sparse activation specifically in the hidden dimensions. Inspired by the work of (Qiu et al., 2024; Dai et al., 2024; Cai et al., 2024a), we conducted an in-depth analysis of the hidden dimensions and designed a novel sparse activation strategy. This innovation opens a new research avenue for sparsely-activated Transformer architectures.

D. Connection With Other Methods

Connection with MoE: *MoHD and MoE enhance model efficiency through distinct dimensions.* While MoE improves memory capacity by sparsely activating experts in the intermediate dimension (Cai et al., 2024c; Dai et al., 2024; Xue et al., 2024a), MoHD optimizes parameter utilization by leveraging hidden dimension sparsity, directly strengthening representational capabilities through hidden dimension expansion, as shown in Figure 9. Experiments (in Table 2) demonstrate that MoHD achieves superior parameter efficiency under identical activation budgets compared to MoE, though it faces unique challenges like activation collapse and information degradation, necessitating tailored design solutions. The two mechanisms are complementary—MoHD focuses on dynamic feature extraction in the hidden dimension, whereas MoE specializes in parallel computation expansion in the intermediate dimension. Their theoretical compatibility enables synergistic integration, offering a hybrid architecture that balances representational depth with computational resource optimization.

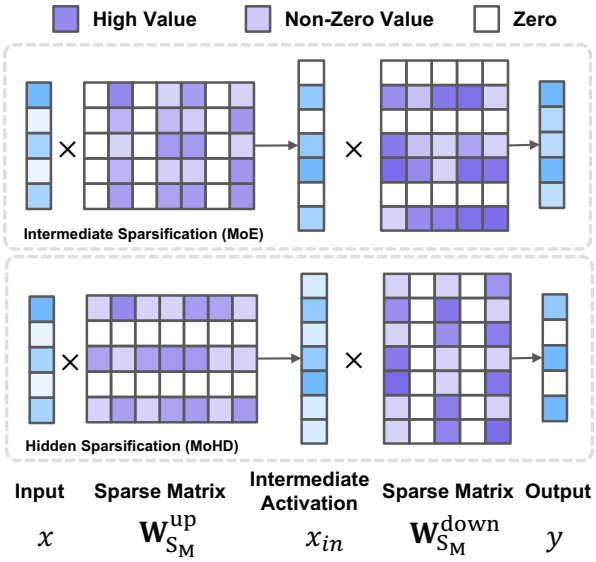


Figure 9. Using FFN as an example, the traditional method (MoE) exploits the sparsity of the intermediate dimension. Our method (MoHD) selectively activates only a subset of hidden dimension parameters across all matrices to enhance efficiency.

Table 5. Theoretical Forward-Pass FLOPS for MoHD Models across Different Configurations (E+12 denotes $\times 10^{12}$)

Model Size	MoHD 50%	MoHD 75%	Baseline 100%	2× Width	3× Width	4× Width
355M	2.70E+12	3.63E+12	4.56E+12	5.40E+12	6.24E+12	7.07E+12
495M	4.19E+12	5.59E+12	7.00E+12	8.40E+12	9.73E+12	1.11E+13
1.13B	6.93E+12	9.80E+12	1.26E+13	1.40E+13	1.55E+13	1.69E+13

Connection with Activation Compression: Some works also prune or quantize hidden dimension activations during inference. However, MoHD *reduces training costs* significantly by activating only a small portion of parameters during training. This provides scalability that these optimization methods lack. Since MoHD *maintains the same architecture for both training and inference*, it avoids performance degradation. Additionally, MoHD can be combined with these methods for further optimization.

E. Computational Cost Analysis of MoHD Models

We provides a detailed analysis of the training and inference computational costs for various MoHD model scales, encompassing both compression and expansion settings. We quantify these costs primarily through theoretical forward-pass Floating Point Operations (FLOPS), comparing them against baseline models, MoE architectures, and other prevalent sparsification methodologies.

As delineated in Table 5, an increase in model width, while leading to a proportional rise in WTE-related FLOPS, only exerts a modest relative impact on total

computational cost. This characteristic is particularly pronounced in deeper MoHD models, where the ratio of activated parameters to total parameters becomes more efficient during scaling. We assert that MoHD's performance enhancements are not merely a consequence of increased WTE size or a broader parameter count. For instance, the 4x-width model exhibits significantly higher FLOPS than its 3x-width counterpart, yet its performance improvement is marginal. Conversely, the 1.13B model reaps greater performance benefits from MoHD than the 495M model, despite its FLOPS-to-baseline ratio being lower. These findings substantiate the claim that MoHD achieves its efficiency predominantly through **structural sparsity** and **expert specialization**, rather than through brute-force scaling. Furthermore, the computational overhead introduced by MoHD's routing and group fusion layers is minimal.

In comparing MoHD with MoE architectures, it is crucial to recognize that these methodologies enhance model efficiency along distinct dimensions. MoE typically improves memory capacity by sparsely activating experts in the intermediate dimension, whereas MoHD optimizes parameter utilization by leveraging sparsity in the hidden dimension, thereby directly augmenting representational capabilities through hidden dimension expansion. Their routing granularities also differ: MoHD routes at the hidden dimension level, enabling the tailoring of token-specific subspace activations, while MoE routes at the expert (subnetwork) level, primarily within the FFN. Moreover, MoHD's applicability spans both Attention and FFN components, in contrast to MoE's conventional confinement to FFN projections. Empirical evidence indicates that MoHD attains superior parameter efficiency under identical activation budgets when compared to MoE. Despite inherent challenges in MoHD's routing across hidden dimensions, which necessitate novel optimization and implementation strategies, MoHD demonstrates notable improvements over MoE when trained from scratch. We suggest that MoHD and MoE are complementary, with MoHD focusing on dynamic feature extraction in the hidden dimension and MoE on parallel computation expansion in the intermediate dimension, implying potential for synergistic integration into more robust and efficient language models.

When juxtaposed with other sparsification techniques such as pruning and quantization, MoHD presents distinct advantages in terms of training costs and architectural consistency. MoHD significantly reduces training expenses by intrinsically learning sparse activations during the training phase, thereby eliminating the need for iterative pruning cycles or post-training fine-tuning procedures characteristic of traditional methods. A key benefit is that MoHD maintains a consistent architecture across both training and inference stages, which inherently mitigates the performance degradation often observed in post-hoc sparsification techniques due to training-inference mismatches. While these optimization methods can be combined with MoHD for further improvements, MoHD, by focusing on activation sparsity in the hidden dimension, achieves better generalization and stability at inference costs comparable to those of activation sparsification approaches.

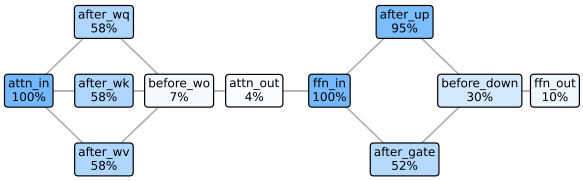


Figure 10. Visualization of activation magnitude in LLaMA2-7B layer 30. In the Transformer, multiple layers show a consistent pattern of activation flow.

F. Extended Observations

In this section, we present several key findings that serve as the foundation for the design of the MoHD approach. In Section F.1, we observe the long-tail effect of hidden dimension activation values and define activation sparsity accordingly. We analyze the sparsity distribution and differences between attention and FFN across different layers. In Section F.2, we analyze activation flow in Transformers, highlighting compression patterns, stabilization by residuals and normalization, and functional layer differences. In Section F.3, we further identify the existence of shared continuous high activation behaviors and unique discrete high activation behaviors across tokens. Finally, we analyze these phenomena and propose motivations for designing feasible hidden dimension sparsification methods.

F.1. Sparsity in Tokens' Hidden Dimension

For a more comprehensive understanding, we observe the activation magnitudes of 4096 hidden dimensions in LLaMA2-7B. As shown in the left panel of Figure 2, we visualize the relationship between dimension magnitudes and reordered dimension indices based on magnitude size.

Similar to previous observations (Liu et al., 2024), the activation of hidden dimensions exhibits a long-tail sparsity phenomenon. For instance, in the input Attention activations of LLaMA2-7B's 16th layer, the cumulative magnitude of the top 1000 dimensions accounts for 71.96% of the total magnitude. In contrast, most dimensions have low activation values, indicating that the model does not utilize information from the majority of hidden dimensions, leading to substantial sparsity in activations.

We also visualized the sparsity of activations in the input and output of Attention and FFN components. Our obser-

vations reveal **significant differences in the magnitude of hidden activations across positions.** Attention exhibits higher activation magnitudes, while FFN activations are comparatively lower. At the input stage, activation magnitudes are relatively high (median > 1), whereas at the output stage, activation magnitudes drop significantly (median < 0.5). The sparsity of hidden dimensions in the input components is consistent across different modules, likely due to the influence of residual connections. However, **at the output stage, the sparsity patterns of Attention and FFN differ markedly.** As shown in the middle panel of Figure 2, Attention demonstrates significant fluctuations in sparsity, with alternating high and low sparsity distributions. In contrast, FFN sparsity remains relatively stable. These differences highlight the distinct functional roles and information processing characteristics of Attention and FFN, prompting us to consider differentiated activation designs for these components.

F.2. Activation Flow in Transformer

We also investigate the variations in activation magnitudes within a single Transformer block, as illustrated in Figure 10. **Consistent activation flow patterns were observed across different Transformer blocks.** The Attention module compresses input activations normalized to 100% through projections (W_Q, W_K, W_V) and weighted averaging, reducing activation magnitudes to 6.7% at W_O. This highlights its ability to suppress irrelevant information through weighted aggregation, while also **showcasing significant functional differences between layers**, as the output activation magnitudes vary to accommodate layer-specific roles. In contrast,**the FFN module demonstrates stable activation patterns**, with compression arising from high-dimensional projections, nonlinearity that sparsifies activations, and dimensionality reduction through linear weighted summation, collectively reducing activation magnitudes.

F.3. Continuous High Activation

We further investigate the temporal correlation of activation sparsity by observing high activation values across different tokens and analyzing the indices that are repeatedly activated by multiple tokens. **A clear correlation in activations is observed over consecutive tokens.** Figure 2 Right shows the number of commonly highly activated dimensions across 2 to 9 consecutive tokens, with the x-axis representing the threshold for defining high activation. When using the top 20% of activation values as the threshold, 2672 dimensions are commonly activated across 2 consecutive tokens, and 673 dimensions remain commonly activated across 9 consecutive tokens.

Figure 3 further illustrates the correlated activation patterns over 5 tokens, where the 4096 hidden dimensions are clustered and reordered based on their activation patterns. Approximately 400 dimensions are commonly highly activated across all 5 tokens, while about 200 dimensions are uniquely highly activated within each token. This indicates that **each token's activations contain shared sub-dimensions that are commonly activated and token-specific sub-dimensions that are independently activated.** Shared high activations model the similarity information shared across tokens in hidden dimensions, while specialized unique activations capture differences. These observations inspired the shared-specialized activation mechanism in the subsequent design of MoHD.

G. More Implementation Details

G.1. MoHD Block Implementation

In Sections 2.1 and 2.2, we observed activation differences across components in various layers, prompting us to design separate routing mechanisms for the Attention and FFN components. Specifically, in one Transformer Block, $\text{Gate}_{\text{attn}}(x, \delta, N, \varphi)$ and $\text{Gate}_{\text{ffn}}(x, \delta, N, \varphi)$ producing scores that determine the activation of dimension-specific sub-dimensions for the output:

$$a, x_s = \text{Gate}_{\text{attn}}(x, \delta, N, \varphi), m, x_s = \text{Gate}_{\text{ffn}}(x, \delta, N, \varphi).$$

In practice, different components may employ distinct sparsification settings. However, for simplicity, we use the same notation throughout this section to represent these settings in a unified manner. Based on the scores from the Router, MoHD applies synchronized sparsification to the hidden dimensions of all up-projection and down-projection matrices, as well as the input x. From Equation 4, we transform $\mathrm{W^Q}, \mathrm{W^K}, \mathrm{W^V}, \mathrm{W^O}, \mathrm{W^{up}}, \mathrm{W^{gate}}, \mathrm{W^{down}}$ into MoHD 's sub-dimensions $\|_{i=1}^{N} \mathrm{Q}_i, \|_{i=1}^{N} \mathrm{K}_i, \|_{i=1}^{N} \mathrm{V}_i, \|_{i=1}^{N} \mathrm{O}_i$ and $\|_{i=1}^{N} \mathrm{UP}_i, \|_{i=1}^{N} \mathrm{GATE}_j i, \|_{i=1}^{N} \mathrm{DOWN}_i$. We substitute these into the sparsified Attention and FFN defined in Equation 13 and 11, yielding outputs y_a and y_m, respectively:

$$\mathrm{MHA_{MoHD}}(x_s) = \mathrm{M_a}\alpha_{\mathrm{a}} \sum_{\mathrm{i=1}}^{\mathrm{h}} \mathrm{Head_i}\left(\|_{\mathrm{j=1}}^{\mathrm{N}} \mathrm{a_j}\, \mathrm{O_j(x_s)}\right), \quad (14)$$

$$\begin{aligned}\mathrm{Head}_i = \sigma\Big(&\Big(x_s\left(\|_{j=1}^{N} a_j\, \mathrm{Q}_j(x_s)\right)\\ &\left(x_s\left(\|_{j=1}^{N} a_j\, \mathrm{K}_j(x_s)\right)\right)^{\top}\Big)\frac{1}{\sqrt{d_h}}\Big)\\ &\times x_s\left(\|_{j=1}^{N} a_j\, \mathrm{V}_j(x_s)\right),\end{aligned} \quad (15)$$

$$\begin{aligned}\mathrm{FFN_{MoHD}}(x_s) = \mathrm{M_m}\alpha_{\mathrm{m}}&\left(\|_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{m_i}\, \mathrm{DOWN_i(x_s)}\right)\\ &\times \sigma\left(x_s\left(\|_{i=1}^{N} m_i\, \mathrm{UP}_i(x_s)\right)\right.\\ &\left.\odot\ x_s\left(\|_{i=1}^{N} m_i\, \mathrm{GATE}_i(x_s)\right)\right).\end{aligned} \quad (16)$$

Table 6. Parameter configurations of MoE under compression and expansion experiments. We use the same settings for only FFN.

MoH	50%	75%	×2	×3	×4
n shared experts	1	1	1	1	1
n routed experts	7	7	7	11	16
num experts per tok	3	5	3	3	3
expert interdemiate ratio	12.5%	12.5%	25%	25%	25%

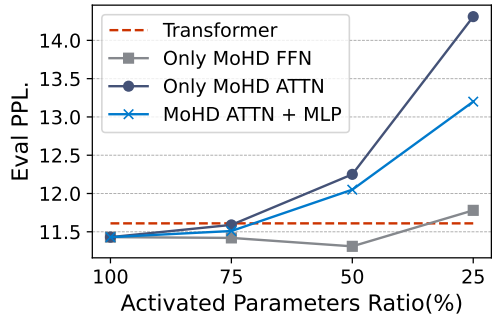


Figure 11. The model's Eval PPL under different sparsity settings applied to Attention and FFN components at varying ratios.

We construct a **MoHD BLOCK** based on MoHD specified MHA and FFN components. Residual connections are designed to further mitigate information loss during the specified forward pass. Following the configuration of LLAMA, we apply LayerNorm layers before the input to both MHA and FFN; however, for simplicity, these are omitted in the formal equations. This process can be formalized as follows:

$$\text{BLOCK}_{\text{MoHD}}(x) = \text{FFN}_{\text{MoHD}}(\text{MHA}_{\text{MoHD}}(x)+x)+x. \quad (17)$$

To train the model effectively, we combine cross-entropy loss $\mathbb{L}_{\text{CE}}$ for language model pre-training and the load balance loss $\mathbb{L}_{\text{B}}$, resulting in the final training objective:

$$\mathbb{L} = \mathbb{L}_{\text{CE}} + \mathbb{L}_{\text{B}}. \quad (18)$$

H. Extended Experiment Details

H.1. Decoupled MoHD Components Setting

To investigate MoHD sparsification, we built three models based on LLaMA2-355M: one with sparsification applied to Attention, one to the FFN, and one to both, as shown in Figure 11. All models were trained on 10B tokens. **# Activate** refers to activated parameters with the total parameter remaining constan. Table 4 presents the effects of decoupled MoHD at 100%, 50%, and 25% sparsity.

MoHD Architecture Advantages. Even with 100% sparsity (where activated parameters match the original model), MoHD outperformed the baseline. This is likely due to its optimized activation dimension allocation and grouped fusion, which suppresses noise from redundant activations and enhances performance.

FFN Exhibits Greater Redundancy. The FFN layer shows more redundancy in hidden dimensions, causing minimal performance loss (and sometimes improvement) when sparsified. In contrast, sparsifying Attention results in a more substantial performance drop. For the 50% sparsity setting, the FFN uses only 195M activated parameters compared to 239M for Attention. This suggests the FFN is better suited for MoHD transformation. The FFN achieved a -0.30 PPL reduction at 50% sparsity, likely due to a decrease in redundant activations that reduces overfitting, whereas Attention sparsification led to a +1.04 PPL increase.

Combining MoHD Attention and FFN Leads to Enhanced Performance. Joint sparsification of Attention and FFN yields the best parameter efficiency. The 50%ATTN-50%FFN model achieved a PPL of 12.05 with only 145M activated parameters—lower than both 50%FFN and 50%ATTN configurations. This joint sparsification outperforms the 25% FFN configuration by 0.19 PPL, because it enhances the consistency of activated hidden dimensions, preserving better learning capacity.

I. Discussion

I.1. Connection with Mixture of Experts Methods

Mixture of Hidden-Dimensions (MoHD) and Mixture of Experts (MoE) are both sparse activation architectures designed to enhance Transformer efficiency, but they operate on fundamentally different dimensions of the model architecture. MoHD focuses on improving the efficiency of scaling the **hidden dimension**, which represents the embedding size of tokens and is related to the model's width or representational capacity per token. In contrast, MoE targets the efficiency of scaling the **intermediate dimension** within the Feed-Forward Networks (FFNs), effectively increasing the model's depth or memory capacity by activating a subset of specialized "experts" or subnetworks.

MoHD achieves efficiency by observing and exploiting the sparsity within the hidden dimension activations. It posits that not all hidden dimensions are equally important for every token, identifying both shared sub-dimensions active across multiple tokens and specialized sub-dimensions unique to individual tokens. By introducing a routing mechanism that dynamically selects and activates only a subset of these hidden sub-dimensions, MoHD reduces the number of activated parameters across all relevant matrices (including those in Attention and FFN layers). This approach allows for the expansion of the total hidden dimension, thereby enhancing the representational power of each token, without a proportional increase in computational cost.

Conversely, MoE architectures enhance efficiency by introducing multiple expert networks in the intermediate dimension of FFN layers. For each token, a gating mechanism selects and activates only a sparse subset of these experts. This effectively expands the total parameter count and memory capacity available to the model by adding more parallel subnetworks, but the activation remains sparse across these experts, reducing the computational cost compared to activating all experts. MoE's primary mechanism is token-conditional routing to different intermediate pathways, typically confined to the FFN module, rather than sparsifying the core hidden dimension across all components.

Thus, the core distinction lies in their dimensionality of focus: MoHD addresses the efficiency of the model's **width** by sparsely activating hidden dimensions to improve token representation, while MoE addresses the efficiency of the model's **depth** or intermediate capacity by sparsely activating experts to enhance computational and memory scalability. Although they operate on different dimensions, these two approaches are complementary and can be considered for synergistic integration to potentially leverage sparsity in both hidden and intermediate spaces.

I.2. Design Principles of Sub-Dimension Routing

The design of the Mixture of Hidden Dimensions (MoHD) architecture, particularly its sub-dimension routing mechanism, is fundamentally driven by empirical observations regarding the activation patterns within Transformer hidden states. Analysis of trained Transformers reveals significant sparsity in the hidden dimensions, where a substantial portion of dimensions exhibit low activation values and contribute minimally to the overall activation magnitude. This long-tail distribution of activation magnitudes indicates that the model does not fully leverage all available dimensions uniformly across all tokens. Consequently, the traditional Transformer approach of treating all hidden dimensions equally results in considerable computational and memory overhead, especially as models scale. This observed sparsity motivates the necessity of a dynamic mechanism that can selectively activate only a relevant subset of hidden dimensions for each input token, thereby improving efficiency by concentrating computational resources on the most informative subspaces rather than processing the entire, often redundant, dimensionality. The dynamic routing mechanism in MoHD serves this purpose by allowing the model to adaptively determine which dimensions are activated based on the specific characteristics of the input, avoiding the computational burden associated with full hidden dimension activation.

Further investigation into the highly activated dimensions across multiple tokens reveals a more nuanced structure than simple universal sparsity. We observe that some dimensions are consistently highly activated across a sequence of tokens, termed **shared sub-dimensions**, while others are uniquely activated for individual tokens, referred to as **specialized sub-dimensions**. This distinction suggests a functional divergence within the hidden space: shared dimensions likely capture common features and representational similarities that are relevant across different tokens, facilitating the processing of general linguistic patterns. In contrast, specialized dimensions appear to be crucial for encoding fine-grained, token-specific semantic differences and higher-level contextual information. This empirical finding directly inspires the structured routing approach in MoHD, which decouples the handling of these two types of dimensions. By partitioning the hidden space into shared and specialized subspaces and employing a routing mechanism that guarantees consistent activation for the shared portion while dynamically selecting from the specialized portion based on token input, MoHD aims to effectively model both the commonalities and unique characteristics of tokens. This mixed activation strategy, balancing universally relevant features with token-specific details, is central to MoHD's ability to enhance representational capacity and parameter efficiency simultaneously.

I.3. Activation Flow Maintenance and Load Balancing

The design of Activation Flow Maintenance in MoHD is motivated by empirical observations of consistent activation flow patterns within Transformer blocks. Specifically, it was noted that Attention modules compress input activations, while FFNs maintain stable activation patterns. However, applying sparse activation to hidden dimensions, as done in MoHD, can lead to information degradation. This degradation arises from the router's softmax-normalized weights, which may cause some dimensions to receive disproportionately high weights while others are neglected. Furthermore, the parallel concatenation of activated sub-dimension outputs can suppress information in low-weighted dimensions without compensation. To counteract these issues and maintain robust activation flow despite sparse activation, MoHD employs several strategies.

One key strategy is Sub-dimension Scaling. This mechanism addresses the suppression of sub-dimension activations caused by softmax weight normalization. A scaling factor, α, is introduced to ensure that the sum of activation weights across all dimensions remains consistent with the input magnitude, allowing activated dimensions to retain their proportional influence and preserving stable activation magnitudes across dimensions.

Additionally, the Grouped Fusion Layer is introduced to mitigate information loss from the sparse output. This layer projects the sparsified hidden-dimension output, y_s, back to the original dimension d. To reduce computational over-

head, a Monarch matrix is used for efficient grouped fusion mapping. Given a receptive field r, this mapping matrix M is structured to perform grouped filling and mapping after sparse activation. This process reconstructs information across the original hidden dimensions while maintaining computational efficiency, preserving information integrity, and improving dimension utilization without significantly increasing parameters.

Beyond maintaining the activation flow within the components, it is crucial to ensure that the learned routing strategy effectively utilizes all available specialized sub-dimensions. Research on conditional computation architectures has shown that automatically learned routing can suffer from load imbalance, where only a few sub-dimensions are frequently selected, leaving others underutilized. To address this and encourage a more even distribution of tokens among different sub-dimensions, the Sub-Dimension Load Balance Loss is incorporated. This loss penalizes imbalances in sub-dimension assignments. By encouraging the gating mechanism to distribute assignments more uniformly across sub-dimensions, the balance loss helps prevent router collapse, where the router consistently selects only a small subset of sub-dimensions, and ultimately leads to improved utilization and overall model efficiency.

I.4. Limitations and Future Research Directions

Despite its demonstrated effectiveness in improving parameter efficiency and task performance, the Mixture of Hidden Dimensions (MoHD) architecture has several existing limitations that present opportunities for future research. A primary challenge lies in the **sensitivity of MoHD to hyperparameters**, specifically the sparsity ratio δ, the proportion of shared sub-dimensions φ, and the total number of sub-dimensions N. Achieving the optimal balance between shared and specialized sub-dimensions is crucial for maximizing generalization and specialization, requiring careful tuning. Furthermore, **optimizing the routing mechanism** poses a significant challenge. Unlike the routing in Mixture of Experts (MoE) models which activate experts in the intermediate dimension, MoHD activates sub-dimensions within a single matrix, which is inherently more complex. While effective, the current routing loss stability decreases as the number of sub-dimensions increases, potentially limiting the scalability of the routing design itself. Another limitation arises from the **growth of the Word Token Embedding (WTE) layer**. As MoHD expands the effective hidden dimension (model width), the WTE layer scales proportionally, contributing a non-negligible portion of total parameters and potentially offsetting some efficiency gains, particularly at very large scales. Finally, despite the inclusion of activation scaling and group fusion mechanisms, there is a risk of **information degradation under extreme sparsity**. Large-scale downsampling and the softmax weighting in the routing can lead to skewed distributions, potentially suppressing useful but low-weighted sub-dimensions and reducing representational fidelity. Addressing these limitations is vital for the broader applicability and further enhancement of MoHD.

Exploring the application of MoHD to **larger-scale Large Language Models (LLMs)** presents significant potential despite associated challenges. Larger models typically exhibit higher parameter redundancy, making MoHD's sparse activation mechanism potentially more effective in reducing inefficiencies without sacrificing expressive power. Experiments on smaller scales indicate that MoHD's performance improvement over baselines increases with total parameter count, suggesting that the benefits may scale positively with model size. By combining shared and specialized sub-dimensions, MoHD enhances the model's capacity to capture both general and fine-grained, token-specific patterns. Scaling up could increase the size of each sub-dimension, improving its ability to model complex language phenomena and potentially enhancing generalization across tasks. However, pretraining larger LLMs is computationally expensive, hindering immediate full-scale experiments. Future work should aim to validate these potential benefits and navigate the associated training complexities.

The core principles of MoHD may also extend beyond Natural Language Processing (NLP) tasks, particularly to other domains that utilize Transformer architectures, such as **Vision-Language Models (VLMs)**. Vision Transformers (ViTs), for instance, process image patches as tokens, similar to text tokens. These visual patches exhibit both global patterns shared across the image and local unique features, analogous to linguistic semantics. If the hidden dimensions in ViTs also demonstrate shared and token-specific activation patterns, the principle of selectively activating sub-dimensions could enhance efficiency and expressiveness in vision tasks. However, such application would require **architectural adaptation** as visual representations differ significantly from language. Tuning routing and partitioning strategies to align with visual characteristics is necessary. Furthermore, in multimodal settings like VLMs, incorporating sparsity-aware mechanisms is crucial to avoid harming the alignment across modalities. Direct transfer is not trivial, but the generalizability of MoHD's principles warrants exploration in vision and multimodal settings.

Future research directions for MoHD should focus on enhancing the robustness and applicability of the architecture. Improving the **stability and scalability of the routing mechanism** is a critical area, particularly as the number of sub-dimensions increases. Investigating alternative routing strategies or loss functions could help address load imbalance issues and prevent router collapse, ensuring all sub-dimensions are actively utilized. Further analysis and

Table 7. Detailed configuration, activation parameters, and total parameters of the models included in our study. L.2-355M represents the LaMMA-2 architecture model with 355M total parameters.

Model Setting	**L.2-355M**	**L.2-495M**	**L.2-1.13B**
hidden size	1024	1536	2048
intermediate size	2560	2560	4096
attention heads	32	32	32
num kv heads	32	16	32
layers	24	24	24
# Activate	289M	396M	1B
# Params	355M	495M	1.13B

Table 8. Parameter configurations of MoHD under compression and expansion experiments. We use the same settings for both Attention and FFN. For detailed reasoning behind these configurations, please refer to Analysis.

MoH	50%	75%	×2	×3	×4
attn top k	4	4	4	4	4
attn sub-dim num	8	12	8	12	16
ffn top k	4	4	4	4	4
ffn sub-dim num	8	12	8	12	16
shared sub-dim num	3	3	3	3	3
group fusion dim	8	12	8	12	16

optimization of the **WTE layer's growth** relative to total computational cost are needed to maximize efficiency gains at very large scales. Exploring the potential for **synergistic integration with other sparsity methods**, such as Mixture of Experts (MoE), could unlock sparsity benefits across multiple dimensions, combining MoHD's focus on hidden dimension dynamism with MoE's intermediate dimension expansion. Additionally, conducting **more extensive ablation studies** on the interaction between shared and specialized sub-dimensions and analyzing the impact of different routing parameters (e.g., K-value selection) would provide deeper insights into MoHD's mechanisms and guide further design improvements. Finally, empirical validation on **larger-scale models and diverse non-NLP tasks** is essential to demonstrate the broader potential and identify specific challenges in different domains.