The Intersection of Modular Architectures and Scalable AI Systems

Yusuf Midha , Harith Husni , Fawzi Gamal *

Posted Date: 5 August 2025

doi: 10.20944/preprints202508.0288.v1

Keywords: Mixture of Experts; sparse activation; conditional computation; expert routing; gating functions; Modular Deep Learning; universal approximation; continual learning; sparse models; Neural Network Architectures

 doi:10.20944/preprints202508.0288.v1

Article

The Intersection of Modular Architectures and Scalable AI Systems

Yusuf Midha, Harith Husni and Fawzi Gamal *

Department of Computer Science and Technology, King Abdullah University of Science and Technology
* Correspondence: fawzi.gamal@kaust.edu.sa

Abstract

Mixture of Experts (MoE) architectures have emerged as a fundamental framework in contemporary deep learning, enabling scalable conditional computation through the dynamic activation of a sparse subset of expert subnetworks. By decoupling capacity from computational cost, MoEs achieve unprecedented parameter efficiency while maintaining or exceeding the predictive performance of dense models. This survey presents an in-depth theoretical and empirical analysis of MoE models, with particular emphasis on their structural properties, functional capacity, and training dynamics. We formally define the general MoE function class as: $f(\mathbf{x}) = \sum_{m=1}^{M} G_m(\mathbf{x}) \cdot E_m(\mathbf{x})$, where E_m are expert networks and G_m are gating coefficients satisfying a sparsity constraint $\|\mathbf{G}(\mathbf{x})\|_0 \leq k \ll M$. We explore the approximation capabilities of MoEs, proving that under mild assumptions on the gating and expert classes, such models form a universal approximator family. Furthermore, we investigate the effective capacity scaling of MoEs, showing that their VC-dimension and Rademacher complexity grow with the number of experts M, while per-example compute remains bounded by k. The survey categorizes MoE designs into hard vs. soft gating, static vs. dynamic routing, and shallow vs. hierarchical expert arrangements, and evaluates their impact on optimization and generalization. We analyze challenges unique to MoEs, including expert collapse, routing instability, and irregular communication overheads. Recent advances such as Switch Transformers, GShard, V-MoE, and Token Routing are reviewed in the context of these challenges. Finally, we articulate open problems and research frontiers, including optimal gating function design, continual learning via expert expansion, modular interpretability, and the theoretical limits of sparse mixture modeling. This survey aims to provide a unified mathematical foundation and future outlook for Mixture of Experts as a scalable, modular paradigm for efficient and adaptive artificial intelligence.

Keywords: Mixture of Experts; sparse activation; conditional computation; expert routing; gating functions; Modular Deep Learning; universal approximation; continual learning; sparse models; Neural Network Architectures

---

1. Introduction

In recent years, the exponential growth of data and the increasing complexity of machine learning tasks have necessitated the development of models that are not only expressive and scalable but also computationally efficient. Among the architectural paradigms that have garnered substantial attention, the *Mixture of Experts* (MoE) framework has emerged as a powerful mechanism for conditional computation, where only a sparse subset of model parameters is activated for a given input. Originally proposed by Jacobs et al. in the early 1990s, MoE models are rooted in the ensemble learning paradigm, yet distinguished by their dynamic, input-dependent routing of computational resources.

Notation Summary

For clarity, we include a brief summary of the primary notations used throughout this survey:

 doi:10.20944/preprints202508.0288.v1

Table 1. Summary of Notation.

Symbol	Meaning
$E_m(\cdot)$	m-th expert function
$G_m(\mathbf{x})$	Gating weight for expert m at input $\mathbf{x}$
$\mathcal{S}_k(\mathbf{x})$	Top-k selected experts for input $\mathbf{x}$
$f_{\text{MoE}}(\mathbf{x})$	Output of the MoE model at input $\mathbf{x}$
M	Total number of experts
k	Number of active experts per input
n_e	Number of parameters per expert
$\mathcal{F}_{\text{MoE}}$	Function class represented by MoE models
$\mathcal{R}_n$	Rademacher complexity
$\mathcal{T}$	Task identifier in continual learning

Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the input space and $\mathcal{Y} \subseteq \mathbb{R}^k$ the output space. Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ with $\mathbf{x}_i \in \mathcal{X}$ and $\mathbf{y}_i \in \mathcal{Y}$, the goal is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ that minimizes some loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$. In the MoE setting, the hypothesis class $\mathcal{H}_{\text{MoE}}$ is constructed as a composition of a gating network $G : \mathcal{X} \to \Delta^{M-1}$ and a set of expert networks $\{E_m\}_{m=1}^M$, where Δ^{M-1} is the $(M-1)$-dimensional probability simplex. Formally, the MoE model computes the output for an input $\mathbf{x} \in \mathcal{X}$ as:

$$f_{\text{MoE}}(\mathbf{x}) = \sum_{m=1}^{M} G_m(\mathbf{x}) E_m(\mathbf{x}), \tag{1}$$

where $G_m(\mathbf{x})$ denotes the m-th component of the gating function $G(\mathbf{x})$, and $E_m : \mathcal{X} \to \mathcal{Y}$ is the m-th expert [1]. The gating function assigns input-dependent weights to each expert, effectively selecting a subset of experts based on relevance. In sparse variants, such as those employed in large-scale language models (e.g., GShard, Switch Transformers), the sum in Equation (1) is replaced with a sparse sum over the top-k experts:

$$f_{\text{MoE}}^{\text{sparse}}(\mathbf{x}) = \sum_{m \in \mathcal{S}_k(\mathbf{x})} G_m(\mathbf{x}) E_m(\mathbf{x}), \tag{2}$$

where $\mathcal{S}_k(\mathbf{x}) \subseteq \{1, \ldots, M\}$ is the set of indices corresponding to the top-k gating weights. The appeal of MoE lies in its theoretical and empirical promise to approximate complex, multimodal mappings while preserving computational efficiency [2]. Theoretically, MoE models relate closely to function spaces defined by piecewise-linear or piecewise-smooth functions. Under mild assumptions, MoE can approximate any measurable function to arbitrary accuracy, a consequence of the universal approximation theorem when the experts are themselves universal approximators (e.g., multilayer perceptrons). In the context of optimization, training MoE models introduces unique challenges, primarily due to the non-differentiability of top-k selection and the high variance in expert utilization. Various techniques, such as load-balancing regularization, straight-through estimators, and auxiliary loss terms, have been introduced to mitigate these issues. Let θ_G and $\theta_E = \{\theta_m\}_{m=1}^M$ denote the parameters of the gating and expert networks, respectively. The optimization objective becomes:

$$\min_{\theta_G, \theta_E} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\mathcal{L}(f_{\text{MoE}}(\mathbf{x}), \mathbf{y}) + \lambda \cdot \mathcal{R}(\theta_G, \theta_E)], \tag{3}$$

where $\mathcal{R}(\cdot)$ denotes a regularization term (e.g., to enforce balanced expert usage), and λ is a tunable hyperparameter [3]. From a probabilistic perspective, MoE can be interpreted as a hierarchical latent variable model. Let $Z \in \{1, \ldots, M\}$ be a latent variable indicating the active expert [4]. Then the conditional likelihood becomes:

$$p(\mathbf{y} \mid \mathbf{x}) = \sum_{m=1}^{M} p(Z = m \mid \mathbf{x}) p(\mathbf{y} \mid \mathbf{x}, Z = m), \tag{4}$$

 doi:10.20944/preprints202508.0288.v1

which aligns with the mixture-of-distributions interpretation, where $p(Z = m \mid \mathbf{x}) = G_m(\mathbf{x})$ and $p(\mathbf{y} \mid \mathbf{x}, Z = m) = E_m(\mathbf{x})$ under deterministic experts [5]. The relevance of MoE in modern AI systems is underscored by its deployment in cutting-edge architectures for natural language processing, vision, and multi-modal learning [6]. As models scale to billions of parameters, the sparse activation in MoE allows for the decoupling of model capacity from inference cost [7]. Yet, this expressivity comes at the cost of architectural complexity, instability during training, and challenges in routing, interpretability, and load balancing. This survey aims to provide a comprehensive and mathematically grounded review of the Mixture of Experts paradigm, encompassing classical formulations, recent advancements in sparse MoE, training algorithms, theoretical underpinnings, and diverse applications across machine learning domains. Through rigorous exposition, we elucidate the principles and design choices that govern MoE models, shedding light on their power, limitations, and future research directions [8].

2. Background and Preliminaries

The Mixture of Experts (MoE) architecture can be viewed through multiple theoretical lenses, including ensemble learning, conditional computation, probabilistic graphical models, and function approximation theory [9]. This section lays the mathematical groundwork necessary for understanding the full breadth of MoE methodologies by formally describing key components such as expert functions, gating mechanisms, sparsity-inducing strategies, and probabilistic interpretations [10].

2.1. Expert Networks

Let us denote the expert set as $\mathcal{E} = \{E_m(\cdot; \theta_m)\}_{m=1}^{M}$, where each $E_m : \mathcal{X} \to \mathcal{Y}$ is a parameterized function representing the m-th expert, and θ_m are its parameters [11]. In practice, each expert may take the form of a neural network, such as a multilayer perceptron (MLP), convolutional network, or transformer block [12]. The choice of expert architecture significantly influences the representational power of the MoE ensemble [13]. Each expert function can be interpreted as a basis function in a functional basis expansion:

$$f(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m(\mathbf{x}) \phi_m(\mathbf{x}), \tag{5}$$

where $\phi_m(\mathbf{x}) := E_m(\mathbf{x})$ and $\alpha_m(\mathbf{x}) := G_m(\mathbf{x})$ are input-dependent coefficients from the gating function [14].

2.2. Gating Mechanism

The gating network $G : \mathcal{X} \to \Delta^{M-1}$ maps an input $\mathbf{x}$ to a discrete probability distribution over experts, where Δ^{M-1} is the $(M-1)$-dimensional probability simplex:

$$\Delta^{M-1} = \left\{ \mathbf{g} \in \mathbb{R}^M \;\middle|\; g_i \geq 0,\ \sum_{i=1}^{M} g_i = 1 \right\}. \tag{6}$$

A typical instantiation of the gating function is a softmax transformation applied to a learned scoring function $s : \mathcal{X} \to \mathbb{R}^M$:

$$G_m(\mathbf{x}) = \frac{\exp(s_m(\mathbf{x}))}{\sum_{j=1}^{M} \exp(s_j(\mathbf{x}))}. \tag{7}$$

In sparse MoE models, the gating function is modified to activate only the top-k experts [15]. Let $\mathcal{S}_k(\mathbf{x})$ denote the index set of the top-k scores $s_m(\mathbf{x})$ [16]. Then, we define a sparse softmax:

$$G_m^{\text{sparse}}(\mathbf{x}) = \begin{cases} \dfrac{\exp(s_m(\mathbf{x}))}{\sum_{j \in \mathcal{S}_k(\mathbf{x})} \exp(s_j(\mathbf{x}))} & \text{if } m \in \mathcal{S}_k(\mathbf{x}), \\ 0 & \text{otherwise[17].} \end{cases} \tag{8}$$

 doi:10.20944/preprints202508.0288.v1

2.3. Probabilistic Interpretation

From a generative modeling standpoint, MoE may be interpreted as a latent variable model with a discrete latent variable $Z \sim \text{Categorical}(G(\mathbf{x}))$, which determines the responsible expert for each input $\mathbf{x}$. The joint distribution can be written as:

$$p(\mathbf{y}, Z = m \mid \mathbf{x}) = G_m(\mathbf{x}) \cdot p(\mathbf{y} \mid \mathbf{x}, Z = m), \tag{9}$$

with the marginal likelihood given by:

$$p(\mathbf{y} \mid \mathbf{x}) = \sum_{m=1}^{M} G_m(\mathbf{x}) \cdot p(\mathbf{y} \mid \mathbf{x}, Z = m). \tag{10}$$

This probabilistic view facilitates the use of Expectation-Maximization (EM) algorithms for training, especially in scenarios where the expert outputs correspond to probabilistic densities or classifiers.

2.4. Function Approximation Perspective

From the perspective of approximation theory, MoE systems can be shown to approximate any continuous function $f \in C(\mathcal{X})$ under mild assumptions [18]. If each expert E_m is drawn from a universal function approximator class (e.g., feedforward neural networks), then for any $\varepsilon > 0$, there exists a gating function G and experts $\{E_m\}$ such that:

$$\sup_{\mathbf{x} \in \mathcal{X}} \left\| f(\mathbf{x}) - \sum_{m=1}^{M} G_m(\mathbf{x}) E_m(\mathbf{x}) \right\| < \varepsilon. \tag{11}$$

This result follows from a generalized Stone-Weierstrass theorem applied to MoE compositions, under the constraint that G and E_m are continuous and suitably bounded [19].

2.5. Load Balancing and Expert Utilization

To avoid underutilization of certain experts, which leads to training inefficiencies and poor generalization, modern MoE models introduce regularization terms that penalize imbalance. Let $\pi_m = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[G_m(\mathbf{x})]$ denote the average usage of expert m [20]. A commonly used load-balancing loss is the entropy regularizer or coefficient-of-variation penalty:

$$\mathcal{R}_{\text{load}} = \lambda \cdot \text{CV}^2(\{\pi_m\}) = \lambda \cdot \frac{\text{Var}[\pi_m]}{(\mathbb{E}[\pi_m])^2}, \tag{12}$$

which encourages uniformity in $\{\pi_m\}_{m=1}^{M}$.

2.6. Training Objectives and Gradient Estimation

Training MoE models requires careful gradient estimation, especially when sparsity leads to discrete selections [21]. Let $\mathcal{L}_{\text{MoE}}(\theta)$ denote the full objective:

$$\mathcal{L}_{\text{MoE}}(\theta) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}}[\ell(f_{\text{MoE}}(\mathbf{x}), \mathbf{y}) + \mathcal{R}_{\text{load}}(\theta)]. \tag{13}$$

In sparse models, straight-through estimators or soft relaxations (e.g., Gumbel-softmax) are employed to approximate gradients through non-differentiable top-k operations. This theoretical background provides the tools necessary to delve into the design, optimization, and practical deployment of MoE architectures [22]. In the subsequent sections, we explore classical and modern variations of MoE, survey optimization techniques, and analyze their empirical performance across multiple domains [23].

 doi:10.20944/preprints202508.0288.v1

3. Taxonomy and Variants of Mixture of Experts

The Mixture of Experts framework admits a rich taxonomy, encompassing numerous architectural variants and operational paradigms. These models vary along multiple axes, including the gating function's structure, expert specialization, sparsity of activation, parameter sharing, and probabilistic versus deterministic routing [24]. In this section, we provide a formal categorization of MoE models, elucidating the distinguishing mathematical properties of each class [25].

3.1. Soft Versus Hard Gating

The most fundamental distinction among MoE models lies in the nature of the gating function $G(\mathbf{x})$.

3.1.1. Soft Gating

In soft-gated MoE models, all experts contribute to the final output with weights proportional to their gating probabilities:

$$f_{\text{soft}}(\mathbf{x}) = \sum_{m=1}^{M} G_m(\mathbf{x}) E_m(\mathbf{x}), \quad G_m(\mathbf{x}) \in [0,1], \quad \sum_{m=1}^{M} G_m(\mathbf{x}) = 1 [26]. \tag{14}$$

This formulation yields a smooth and differentiable function with respect to both the gating parameters and expert parameters. Soft MoE is closely related to attention mechanisms and kernel mixtures in probabilistic models.

3.1.2. Hard Gating

Hard-gated MoE activates a single expert per input, typically chosen as:

$$m^* = \arg\max_m G_m(\mathbf{x}), \quad f_{\text{hard}}(\mathbf{x}) = E_{m^*}(\mathbf{x}) [27]. \tag{15}$$

Since the arg max operator is non-differentiable, hard MoE models often employ surrogate gradient techniques (e.g., REINFORCE, Gumbel-softmax) for training [28].

3.2. Sparse Gating and Top-k Experts

A middle ground between soft and hard MoE is achieved via *sparse gating*, where only a subset $\mathcal{S}_k(\mathbf{x}) \subset \{1, \ldots, M\}$ of k experts are activated [29]. The output becomes:

$$f_{\text{sparse}}(\mathbf{x}) = \sum_{m \in \mathcal{S}_k(\mathbf{x})} G_m(\mathbf{x}) E_m(\mathbf{x}). \tag{16}$$

Sparse MoEs reduce computational cost from $\mathcal{O}(M)$ to $\mathcal{O}(k)$ and are amenable to scalable deployment in large neural architectures such as Switch Transformers and GShard.

3.3. Independent Versus Shared Experts

MoE models may be further classified by whether experts are parameterized independently or share parts of their architecture:

- **Independent Experts:** Each $E_m(\cdot; \theta_m)$ is trained separately and has a distinct parameter set [30]. This offers high specialization but incurs greater memory cost [31].
- **Shared Experts:** Experts may share a common backbone or parameter subsets [32]. For example, each expert may be implemented as a residual transformation over a shared encoder:

$$E_m(\mathbf{x}) = h(\mathbf{x}) + \Delta_m(\mathbf{x}), \tag{17}$$

where $h(\mathbf{x})$ is a shared base and Δ_m is the expert-specific residual.

 doi:10.20944/preprints202508.0288.v1

3.4. Static Versus Dynamic Experts

In static MoE, the expert functions E_m are fixed post-training or specialized via pre-defined tasks [33]. In contrast, dynamic MoE adapts expert parameters or selection policy during training or inference [34]. Some dynamic models even incorporate meta-learning components, where experts are selected based on learned context embeddings [35].

3.5. Hierarchical Mixture of Experts

Hierarchical MoE introduces multiple levels of gating, leading to tree-structured architectures [36]. A typical two-level hierarchical MoE can be expressed as:

$$f(\mathbf{x}) = \sum_{i=1}^{M_1} G_i^{(1)}(\mathbf{x}) \left[\sum_{j=1}^{M_2} G_{i,j}^{(2)}(\mathbf{x}) E_{i,j}(\mathbf{x}) \right], \tag{18}$$

where $G^{(1)}$ is a coarse-grained gate selecting among M_1 super-experts, each of which controls a second-layer MoE with M_2 sub-experts. Hierarchical MoE enables scalable routing with logarithmic complexity in the number of total experts.

3.6. Probabilistic Versus Deterministic Routing

In probabilistic MoE, routing is stochastic and often accompanied by latent variable inference. In deterministic MoE, expert selection is deterministic and often implemented via thresholding or argmax selection. The probabilistic formulation is more amenable to Bayesian learning, while deterministic routing is typically more efficient for deployment.

3.7. Multi-Task and Multi-Modal Mixture of Experts

MoE is particularly well-suited to multi-task learning (MTL) and multi-modal fusion, where each expert is specialized to a task or modality. Let $\mathcal{T} = \{1, \ldots, T\}$ denote the task index space. For multi-task MoE, the gating function may be conditioned on both input $\mathbf{x}$ and task label $t \in \mathcal{T}$:

$$G_m(\mathbf{x}, t) = \frac{\exp(s_m(\mathbf{x}, t))}{\sum_{j=1}^{M} \exp(s_j(\mathbf{x}, t))} [37]. \tag{19}$$

Similarly, in multi-modal MoE, experts are specialized to different input domains (e.g., vision, text, audio), and the gating function aggregates modality-specific signals.

3.8. Notable Architectures

Prominent implementations of MoE include:

- **GShard MoE** [38]: A large-scale MoE with sparse activation and gradient-based load balancing [39].
- **Switch Transformer** [1]: Simplifies MoE by activating a single expert per layer, leading to computational savings and robust scaling [40].
- **BASE Layers** [38]: Balance-efficient MoEs trained with auxiliary losses to encourage uniform routing [41].
- **Task MoE / Multi-gate MoE**: Used in multi-task learning where separate gating networks per task provide flexibility in expert sharing [42].

This taxonomy underscores the diversity of MoE formulations, each making unique trade-offs between expressivity, efficiency, interpretability, and scalability. In the next section, we delve into the optimization techniques employed for training these architectures efficiently and effectively.

4. Training and Optimization Techniques

Training Mixture of Experts (MoE) models introduces several challenges that extend beyond standard neural network optimization [43]. These include (i) gradient estimation under discrete expert

 doi:10.20944/preprints202508.0288.v1

selection, (ii) load balancing to ensure expert utilization, (iii) routing instability, and (iv) large-scale parallelization. In this section, we present a formal treatment of training objectives, regularization methods, sparse gating optimization, and gradient approximation techniques [44].

4.1. Optimization Objective

Let $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N}$ be the dataset, and denote the MoE output as:

$$f_{\text{MoE}}(\mathbf{x}) = \sum_{m \in \mathcal{S}_k(\mathbf{x})} G_m(\mathbf{x}) E_m(\mathbf{x}). \tag{20}$$

The total loss consists of two components: (1) the primary prediction loss $\ell(f_{\text{MoE}}(\mathbf{x}), \mathbf{y})$, and (2) a regularization term $\mathcal{R}_{\text{MoE}}$:

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\ell(f_{\text{MoE}}(\mathbf{x}), \mathbf{y})] + \mathcal{R}_{\text{MoE}}. \tag{21}$$

The regularization term may include load balancing losses, sparsity penalties, entropy constraints, and auxiliary objectives designed to stabilize expert routing [45].

4.2. Load Balancing and Auxiliary Losses

To promote balanced expert usage, many MoE models incorporate auxiliary losses [46]. Let $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[G_m(\mathbf{x})]$ denote the expected activation of expert m [47]. The following are common regularization schemes:

Load Balance Loss:

$$\mathcal{R}_{\text{load}} = \lambda \cdot \left(\frac{\sum_{m=1}^{M} \pi_m^2}{\left(\sum_{m=1}^{M} \pi_m\right)^2} \right), \quad \pi_m := \sum_{i=1}^{N} G_m(\mathbf{x}^{(i)}). \tag{22}$$

This regularizer penalizes high variance in expert usage.

Entropy-Based Regularization:

$$\mathcal{R}_{\text{ent}} = -\lambda \cdot \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \left[\sum_{m=1}^{M} G_m(\mathbf{x}) \log G_m(\mathbf{x}) \right]. \tag{23}$$

This encourages higher entropy in expert selection, promoting exploration and utilization diversity [48].

4.3. Gradient Estimation under Sparse Routing

The sparse selection of experts introduces non-differentiability into the forward path, particularly with the top-k selection operator:

$$\mathcal{S}_k(\mathbf{x}) = \text{TopK}\left(\{s_m(\mathbf{x})\}_{m=1}^{M} \right) [49]. \tag{24}$$

To handle this, several surrogate techniques are employed:

4.3.1. Straight-Through Estimator (STE)

The STE approximates the gradient of a discrete operator by treating it as an identity in the backward pass:

$$\frac{\partial \text{TopK}(s)}{\partial s} \approx \mathbb{I}_{m \in \mathcal{S}_k(s)} [50]. \tag{25}$$

While biased, this estimator is simple and empirically effective [51].

 doi:10.20944/preprints202508.0288.v1

4.3.2. Gumbel-Softmax Relaxation

Let $g_m \sim \text{Gumbel}(0,1)$ be i.i.d. noise samples [52]. The top-k can be approximated via:

$$\tilde{G}_m(\mathbf{x}) = \frac{\exp\big((s_m(\mathbf{x}) + g_m)/\tau\big)}{\sum_{j \in \mathcal{S}_k(\mathbf{x})} \exp\big((s_j(\mathbf{x}) + g_j)/\tau\big)}, \tag{26}$$

where $\tau > 0$ is a temperature parameter. As $\tau \to 0$, the distribution becomes increasingly sparse [53].

4.4. Backpropagation Through MoE

In the case of soft gating, the model remains fully differentiable, and standard backpropagation can be used:

$$\frac{\partial \mathcal{L}}{\partial \theta_m} = \sum_{i=1}^{N} \frac{\partial \ell}{\partial f} \cdot \left(G_m(\mathbf{x}^{(i)}) \cdot \frac{\partial E_m(\mathbf{x}^{(i)})}{\partial \theta_m} \right). \tag{27}$$

For sparse or hard gating, gradient paths are truncated or approximated using the aforementioned techniques [54].

4.5. Expert Routing Instability

Stochasticity in gating can lead to training instability, particularly in early phases. Techniques to stabilize routing include:

- **Temperature annealing** in softmax or Gumbel-softmax to gradually sharpen selection [55].
- **Moving-average smoothing** of gating scores over mini-batches [56].
- **Noise regularization** to encourage robustness in expert activation under input perturbation.

4.6. Parallelization and Scalability

Large-scale MoE models with thousands of experts require distributed training strategies:

Expert Parallelism:

Experts are partitioned across multiple devices. For input $\mathbf{x}$, only the top-k selected experts are activated and routed to relevant devices. This reduces communication overhead.

All-to-All Communication:

High-performance MoE implementations use custom kernels for collective communication, where each device exchanges selected tokens with the devices hosting their top experts.

Token Sharding and Grouped Routing:

For efficiency, tokens are grouped and routed collectively to minimize inter-device traffic and improve memory coalescence [57].

4.7. Convergence Analysis

While convergence guarantees for MoE remain under active research, empirical evidence suggests that sparse MoE training exhibits convergence behavior comparable to dense models when auxiliary losses are properly tuned. Let Θ denote the parameter space and assume bounded gradients:

$$\|\nabla_\theta \mathcal{L}(\theta)\| \leq G, \quad \forall \theta \in \Theta. \tag{28}$$

Then, under a learning rate schedule satisfying $\sum_t \eta_t = \infty$, $\sum_t \eta_t^2 < \infty$, stochastic MoE training with appropriate smoothing converges to a local optimum. This concludes our exploration of training techniques for MoE models. In the next section, we will analyze the empirical performance and applications of these models across a wide range of tasks, from natural language processing to vision and multimodal learning [58].

 doi:10.20944/preprints202508.0288.v1

5. Empirical Performance and Applications

Mixture of Experts (MoE) models have demonstrated remarkable empirical performance across a wide spectrum of tasks, particularly in large-scale settings [59]. Their ability to dynamically allocate model capacity conditioned on input data has enabled significant improvements in parameter efficiency, generalization, and computational scalability. In this section, we survey and analyze the empirical behavior of MoE architectures, benchmarking their performance across multiple domains, and elucidating their task-specific adaptations and benefits [60].

5.1. Evaluation Metrics

The performance of MoE models is typically evaluated via standard predictive metrics, augmented with sparsity-aware and efficiency-specific criteria. Let $\hat{y}^{(i)} = f_{\text{MoE}}(\mathbf{x}^{(i)})$ denote the model's prediction and $y^{(i)}$ the target output. Common evaluation metrics include:

- **Predictive Accuracy:** Classification or regression error over validation set $\mathcal{D}_{\text{val}}$:

$$\text{Acc} = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{val}}} \mathbb{I}\{\arg\max \hat{y} = y\}. \tag{29}$$

- **Perplexity:** For language modeling:

$$\text{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(y^{(i)}|\mathbf{x}^{(i)})\right). \tag{30}$$

- **Expert Utilization Entropy:** Measures uniformity of expert usage:

$$\mathcal{H}_{\text{expert}} = -\sum_{m=1}^{M} p_m \log p_m, \quad p_m = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{m \in \mathcal{S}_k(\mathbf{x}^{(i)})\}[61]. \tag{31}$$

- **Floating Point Operations (FLOPs):** To compare compute-efficiency:

$$\text{FLOPs}_{\text{MoE}} = \sum_{i=1}^{N} \sum_{m\in\mathcal{S}_k(\mathbf{x}^{(i)})} \text{FLOPs}(E_m(\mathbf{x}^{(i)})). \tag{32}$$

5.2. Natural Language Processing (NLP)

MoE models have been most prominently applied in NLP, particularly in transformer-based architectures. Consider a Transformer block where the feedforward layer is replaced by a sparse MoE:

$$\text{FFN}_{\text{MoE}}(\mathbf{x}) = \sum_{m\in\mathcal{S}_k(\mathbf{x})} G_m(\mathbf{x}) \cdot \text{ReLU}(W_m^{(2)} \cdot \text{ReLU}(W_m^{(1)} \cdot \mathbf{x} + b_m^{(1)}) + b_m^{(2)}). \tag{33}$$

Empirical benchmarks demonstrate:

- **Switch Transformer (2022)** [1]: Achieved a 7× gain in training speed with comparable perplexity to dense models.
- **GShard (2021)** [38]: Trained 600B-parameter models on multilingual translation tasks with superior BLEU scores.
- **Task-MoE (2023)**: Enabled parameter-efficient multi-task learning with dynamic expert routing [62].

 doi:10.20944/preprints202508.0288.v1

5.3. Vision Applications

In computer vision, MoE has been integrated into convolutional and vision transformer architectures [63]. For example, in Vision MoE (V-MoE) [64], the MoE block is inserted between attention layers:

$$\text{MoE-MLP}(\mathbf{x}) = \sum_{m \in \mathcal{S}_k(\mathbf{x})} G_m(\mathbf{x}) \cdot \phi_m(\mathbf{x}), \tag{34}$$

where ϕ_m is an expert-specific multi-layer perceptron. Key findings:

- MoE ViT achieves state-of-the-art top-1 ImageNet accuracy with 4× fewer FLOPs than dense ViT [65].
- Experts tend to specialize on image patches of specific geometric or semantic characteristics [66].

5.4. Multimodal Learning

MoE has shown promise in multimodal architectures, where each expert is specialized to a modality (e.g., text, image, audio) or cross-modal interaction [67]. In a multimodal MoE model, let $\mathbf{x} = [\mathbf{x}^{(v)}, \mathbf{x}^{(t)}]$ be the concatenation of visual and textual embeddings [35,68]. The gating function can be factorized:

$$G_m(\mathbf{x}) = \sigma\left(\alpha_m^{(v)} \cdot f_v(\mathbf{x}^{(v)}) + \alpha_m^{(t)} \cdot f_t(\mathbf{x}^{(t)})\right), \tag{35}$$

allowing adaptive expert routing based on modality salience. Applications include:

- **CLIP-MoE**: Specializes experts on alignment between text and vision.
- **VATT-MoE**: Enhances video-audio-text embeddings via dynamic expert routing [69].

5.5. Few-Shot and Transfer Learning

MoE models exhibit strong few-shot generalization [70]. Experts trained on different tasks or domains can be selectively reused:

$$\text{Transfer}(f_{\text{MoE}}) = \sum_{m \in \mathcal{S}_k(\mathbf{x};\text{new})} G_m^{\text{new}}(\mathbf{x}) E_m^{\text{pretrained}}(\mathbf{x}). \tag{36}$$

MoE enables modular transfer by freezing experts and learning new gating functions.

5.6. Ablation and Scaling Studies

Empirical studies further explore the influence of various factors:

- Increasing M (number of experts) improves capacity but may cause routing collapse without regularization.
- Larger k increases compute but smooths gradients and improves stability [71].
- MoE achieves Pareto optimal trade-offs between FLOPs and accuracy in many settings [72].

5.7. Limitations in Empirical Use

Despite their empirical strengths, MoE models face practical issues:

- High variance in expert usage without load balancing.
- Routing collapse where a small subset of experts dominate.
- Communication overhead in distributed setups.
- Difficulty in debugging due to implicit specialization [73].

In summary, MoE models achieve superior empirical performance across domains, especially when compute cost is decoupled from model size. The next section will address the theoretical properties and open questions regarding expressivity, approximation bounds, and generalization of MoE architectures.

 doi:10.20944/preprints202508.0288.v1

6. Theoretical Properties and Expressivity

Mixture of Experts (MoE) architectures exhibit unique theoretical properties arising from their conditional computation and modular structure [74]. In this section, we explore the expressive capacity, approximation bounds, and generalization behavior of MoE models. We further analyze their potential for modular composition, universality, and implicit regularization under sparse expert activation.

6.1. Universal Approximation Properties

Let $\mathcal{F}_{\text{MoE}}$ denote the class of functions realizable by an MoE model with M experts and k-sparse gating:

$$\mathcal{F}_{\text{MoE}} = \left\{ f(\mathbf{x}) = \sum_{m \in \mathcal{S}_k(\mathbf{x})} G_m(\mathbf{x}) E_m(\mathbf{x}) \ : \ G \in \mathcal{G}, E_m \in \mathcal{H} \right\}, \tag{37}$$

where $\mathcal{G}$ is the set of gating functions and $\mathcal{H}$ the hypothesis class for experts [75].

Proposition 1 (Universal Approximation):

If each expert E_m is a universal approximator (e.g., a sufficiently wide feedforward neural network), and if the gating function can select any subset $\mathcal{S}_k$ conditioned on $\mathbf{x}$, then $\mathcal{F}_{\text{MoE}}$ is a universal approximator. *Proof Sketch:* For any function $f^* \in C(\mathbb{R}^d)$ and $\epsilon > 0$, partition the input space into M regions $\{\Omega_m\}$ such that f^* is ϵ-close to a simple function f_m on Ω_m [76]. Set $G_m(\mathbf{x}) \approx \mathbb{I}_{\mathbf{x} \in \Omega_m}$ and $E_m(\mathbf{x}) \approx f_m(\mathbf{x})$. Then f_{MoE} approximates f^* with error at most ϵ [77].

6.2. Capacity Scaling with Experts

Unlike dense networks, MoE models scale their representational power with the number of experts M without proportionally increasing compute, due to sparse activation. Denote by $\mathcal{N}_{\text{dense}}(n)$ a dense network with n parameters, and $\mathcal{N}_{\text{MoE}}(M, k)$ an MoE with M experts, each of size n_e, and k-sparse selection:

$$\text{Total Parameters: } \Theta(M \cdot n_e), \quad \text{Active Parameters per Example: } \Theta(k \cdot n_e).$$

Theorem 1 (Exponential Gain in Capacity):

Suppose each expert E_m belongs to a function class with VC-dimension d_e, and $k \ll M$. Then the MoE model class has effective capacity:

$$\text{VC}(\mathcal{F}_{\text{MoE}}) \geq \Omega(M \cdot d_e), \quad \text{while only } O(k \cdot d_e) \text{ parameters are used per input.} \tag{38}$$

6.3. Modular Representations and Disentanglement

MoE models encourage modular representations by allowing different experts to specialize on distinct input subspaces. Let $\mathbf{x} \in \mathcal{X}$ and assume a latent decomposition $\mathbf{x} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(L)})$ [78]. Then MoE can learn a factorized representation:

$$f(\mathbf{x}) = \sum_{m=1}^{M} G_m(\mathbf{x}) \cdot E_m(\mathbf{x}^{(i_m)}), \tag{39}$$

where each expert E_m focuses on a particular latent factor $\mathbf{x}^{(i_m)}$. This supports implicit disentanglement without explicit supervision [79].

6.4. Generalization Under Sparse Activation

Generalization bounds for MoE models derive from capacity control via sparsity and modularity [80]. Suppose a model activates at most k out of M experts per input, with each expert in a hypothesis class $\mathcal{H}$ of Rademacher complexity $\mathcal{R}_n(\mathcal{H})$. Then the overall Rademacher complexity satisfies:

$$\mathcal{R}_n(\mathcal{F}_{\text{MoE}}) \leq k \cdot \mathcal{R}_n(\mathcal{H}) + \mathcal{R}_n(\mathcal{G}), \tag{40}$$

 doi:10.20944/preprints202508.0288.v1

where $\mathcal{R}_n(\mathcal{G})$ accounts for the gating function complexity [81].

Implication:

MoE models can achieve high expressivity while maintaining generalization, provided that the number of active experts k is small and the gating function is regularized [82].

6.5. Expressivity via Piecewise Function Composition

MoE models define piecewise functions over the input space. Consider a top-1 MoE model:

$$f(\mathbf{x}) = E_{m^*(\mathbf{x})}(\mathbf{x}), \quad m^*(\mathbf{x}) = \arg\max_m s_m(\mathbf{x}). \tag{41}$$

Then f is piecewise continuous with regions defined by:

$$\mathcal{R}_m = \left\{\mathbf{x} \in \mathbb{R}^d : s_m(\mathbf{x}) > s_{m'}(\mathbf{x}), \forall m' \neq m\right\}. \tag{42}$$

Within each region, the function is defined by a single expert [83]. This forms a partitioned decision surface, enabling MoE models to emulate decision trees, rule lists, and other hierarchical structures.

6.6. Theoretical Challenges and Open Questions

Despite their expressive power, several theoretical aspects of MoE remain open:

- **Learnability**: Under what conditions can the gating and expert functions converge jointly to a global optimum?
- **Approximation Limits**: What are the lower bounds on approximation error with fixed k and M?
- **Overfitting Risks**: Can MoE models overfit due to implicit overparameterization, despite sparsity at inference time?
- **Compositionality**: Can MoE be used to construct compositional programs with guaranteed semantics [84]?

These questions highlight the need for deeper theoretical investigations into MoE models [85]. In the following section, we discuss recent advances, current limitations, and promising future directions for developing more efficient, robust, and interpretable mixtures of experts.

7. Future Directions and Open Problems

Despite their empirical success and promising theoretical foundations, Mixture of Experts (MoE) models present several open challenges and opportunities for further research. These span algorithmic, theoretical, and practical dimensions [86]. In this section, we outline key future directions, conjectures, and open problems that could substantially advance the field.

7.1. Learning Optimal Gating Functions

One of the most critical components of MoE architectures is the gating function $G_m(\mathbf{x})$, which governs the sparsity pattern and expert selection. While current designs rely on simple softmax-based or top-k heuristics, an optimal gating function remains elusive.

Open Problem 1 (Gating Optimality):

Let $\mathcal{S}_k(\mathbf{x}) = \arg\max_{S\subseteq[M],|S|=k} \sum_{m\in S} s_m(\mathbf{x})$ be the current selection scheme [87]. Define an oracle selector:

$$\mathcal{S}_k^*(\mathbf{x}) = \arg\min_{S\subseteq[M],|S|=k} \|f(\mathbf{x}) - \sum_{m\in S} E_m(\mathbf{x})\|_2^2. \tag{43}$$

Can a learnable approximation to $\mathcal{S}_k^*$ be developed that balances predictive performance and computational complexity?

 doi:10.20944/preprints202508.0288.v1

7.2. Expert Specialization and Diversity Metrics

Understanding and measuring the diversity of expert specialization is key to improving MoE interpretability and robustness [88]. Let $\phi_m(\mathbf{x})$ denote an embedding learned by expert E_m. Define a diversity metric:

$$D_{\text{inter}} = \frac{1}{M(M-1)} \sum_{m \neq m'} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[\text{sim}(\phi_m(\mathbf{x}), \phi_{m'}(\mathbf{x}))], \tag{44}$$

where $\text{sim}(\cdot, \cdot)$ is a similarity function (e.g., cosine).

Open Problem 2 (Specialization Entropy):

How can we regularize training to explicitly minimize D_{inter} while preserving model accuracy? What is the theoretical link between diversity and generalization error?

7.3. Dynamic Routing with Reinforcement Learning and Meta-Learning

Current routing schemes are static and differentiable. However, dynamic routing via reinforcement learning (RL) or meta-learning could potentially lead to more adaptive and task-specific expert allocation.

Research Direction:

Formulate the expert selection process as a Markov Decision Process (MDP):

$$\pi : \mathcal{X} \to \mathcal{P}(\{1, \ldots, M\}^k), \quad \text{maximize } \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[R(\pi(\mathbf{x}))],$$

where R is a reward function capturing prediction quality and computation cost.

7.4. MoE in Continual and Lifelong Learning

MoE models offer a natural structure for continual learning, where new experts can be added over time and old ones frozen.

Open Problem 3 (Catastrophic Forgetting Mitigation):

Can we design gating mechanisms such that new tasks automatically route to new experts, thereby minimizing interference with old tasks? Let $\mathcal{T}_1, \mathcal{T}_2$ be two tasks, and define interference as:

$$\Delta_{\text{task}} = \mathbb{E}_{\mathbf{x} \in \mathcal{T}_1} \left[\| f^{\text{pre}}_{\text{MoE}}(\mathbf{x}) - f^{\text{post}}_{\text{MoE}}(\mathbf{x}) \| \right] [89].$$

How can we guarantee $\Delta_{\text{task}} \approx 0$ as new tasks are introduced [90]?

7.5. Theoretical Limits and Expressivity Gaps

While MoE models are universal approximators, it remains unclear what classes of functions they represent more efficiently than dense models.

Open Problem 4 (MoE Efficiency Hierarchy):

For which function classes $\mathcal{F}$ does there exist a MoE $f_{\text{MoE}} \in \mathcal{F}_{\text{MoE}}$ such that:

$$\| f_{\text{MoE}} - f^* \| \leq \epsilon \text{ with } O(k \cdot n) \text{ active parameters}, \tag{45}$$

but any dense network f_{dense} satisfying the same approximation requires $\Omega(M \cdot n)$ parameters [91]?

7.6. Scalability and Hardware Efficiency

Sparse expert routing poses significant challenges for hardware efficiency due to non-uniform memory access and communication overheads.

 doi:10.20944/preprints202508.0288.v1

Research Challenge:

Design scheduling and expert placement strategies across devices such that:

$$\text{Total Latency} = \max_{i \in \text{device}} \sum_{m \in \mathcal{E}_i} \text{Time}(E_m(\mathbf{x})), \tag{46}$$

is minimized under memory and bandwidth constraints [92].

7.7. MoE in Structured Prediction and Probabilistic Inference

An underexplored direction is the use of MoE models in structured prediction tasks, such as parsing, alignment, or probabilistic inference in graphical models [56,93]. Experts can be aligned with specific structural components.

Open Question:

Can MoE architectures emulate tractable probabilistic inference (e.g., sum-product networks) by routing to locally conditioned experts?

7.8. Interpretable and Modular AI Systems

MoE models open the door to more interpretable, modular neural systems, where each expert has a dedicated semantic function [94].

Future Work:

Define a formal logic or grammar over expert compositions, and build verifiable MoE systems for safety-critical applications such as medical diagnostics or autonomous driving.

7.9. Towards Theoretical Foundations for Mixture Sparsity

Lastly, a foundational theory for mixture sparsity is lacking [95]. Given an expert set $\{E_1, \ldots, E_M\}$ and a sparsity constraint $k \ll M$, what is the optimal trade-off between model depth, sparsity, and generalization?

Conjecture (Sparse Mixture Efficiency):

There exists a sparsity threshold k^* such that:

$$k < k^* \Rightarrow \text{Underfitting}, \quad k > k^* \Rightarrow \text{Overfitting}, \quad k = k^* \Rightarrow \text{Generalization Optimal}. \tag{47}$$

Deriving such a threshold theoretically or empirically remains a major open problem [96].

7.10. Summary of Research Directions

To conclude, the future of MoE research lies in solving the following grand challenges:

1. Learning and regularizing optimal gating functions.
2. Ensuring expert diversity and modular generalization.
3. Incorporating reinforcement and meta-learning in routing [97].
4. Enabling continual and lifelong learning with minimal forgetting [98].
5. Developing theoretical foundations of mixture sparsity and compositionality [99].
6. Aligning MoE with hardware constraints for scalable deployment.
7. Building interpretable and verifiable MoE systems [100].

These directions not only promise to improve MoE performance, but also pave the way toward more adaptive, scalable, and principled artificial intelligence systems [101].

8. Conclusion

Mixture of Experts (MoE) models represent a powerful and flexible paradigm in contemporary artificial intelligence, offering a unique confluence of modularity, conditional computation, and scal-

 doi:10.20944/preprints202508.0288.v1

able expressivity [102]. Through the strategic combination of multiple expert subnetworks activated selectively by input-dependent gating functions, MoE architectures have achieved state-of-the-art results across a range of domains, including language modeling, vision, reinforcement learning, and multimodal learning. In this survey, we have provided a comprehensive and mathematically rigorous analysis of MoE systems. We began by formalizing the core MoE framework, highlighting the essential components — experts, gating functions, and aggregation mechanisms — and elucidating their interaction [103]. We then classified existing MoE designs into hard versus soft gating, sparse versus dense activation, and hierarchical versus flat topologies [104]. This taxonomy provided a foundation for understanding the structural and algorithmic variants currently employed in practice [105]. We explored the theoretical underpinnings of MoE, demonstrating their universal approximation properties, capacity scaling laws, generalization bounds under sparsity, and the modular disentanglement of latent factors. We also analyzed the expressivity of MoE as a piecewise function approximator, revealing its connection to decision trees and conditional computation graphs [106]. These properties underscore MoE's ability to efficiently represent high-complexity functions while maintaining parameter sparsity at inference time. A critical examination of training methodologies — including noisy gating, load balancing, auxiliary losses, routing regularization, and gradient sparsity — revealed both strengths and bottlenecks. Empirical instabilities, communication overhead, expert collapse, and sensitivity to hyperparameters remain significant challenges [107]. Nonetheless, these techniques have enabled the deployment of large-scale MoE systems like GShard, Switch Transformer, and V-MoE, which exhibit remarkable scalability and performance. Looking forward, we identified a suite of open problems and promising research directions [108]. These include designing optimal gating policies, improving expert diversity and interpretability, integrating MoE with reinforcement learning and meta-learning, and leveraging MoE architectures for continual learning, modular reasoning, and structured prediction [109]. Additionally, foundational questions about mixture sparsity, task interference, and the interplay between routing complexity and generalization capacity demand deeper investigation [110]. Mixture of Experts models, situated at the intersection of deep learning, modular design, and conditional computation, offer a fertile ground for innovation [111]. As research continues to advance, MoE architectures may serve as a cornerstone for building more adaptive, compositional, and efficient AI systems — potentially guiding the development of intelligent agents that reason over modular structures, dynamically allocate resources, and generalize across diverse tasks and environments.

Final Remarks

As we stand on the frontier of scalable and interpretable machine learning, Mixture of Experts offers not only a powerful tool for performance but also a lens into more modular, structured, and cognitively inspired learning systems. Continued theoretical development, rigorous empirical benchmarking, and responsible system design will be critical to unlocking the full potential of MoE in future AI.

References

1. Fedus, W.; Zoph, B.; Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research* **2022**, *23*, 1–39.
2. Zhou, Y.; Du, N.; Huang, Y.; Peng, D.; Lan, C.; Huang, D.; Shakeri, S.; So, D.; Dai, A.M.; Lu, Y.; et al. Brainformers: Trading simplicity for efficiency. In Proceedings of the International Conference on Machine Learning. PMLR, 2023, pp. 42531–42542.
3. Zoph, B.; Bello, I.; Kumar, S.; Du, N.; Huang, Y.; Dean, J.; Shazeer, N.; Fedus, W. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906* **2022**.
4. Ainslie, J.; Lee-Thorp, J.; de Jong, M.; Zemlyanskiy, Y.; Lebron, F.; Sanghai, S. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 4895–4901.
5. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* **2022**.

 doi:10.20944/preprints202508.0288.v1

6. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* **2015**.
7. Qiu, Z.; Huang, Z.; Fu, J. Unlocking Emergent Modularity in Large Language Models. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 2638–2660.
8. Rajbhandari, S.; Li, C.; Yao, Z.; Zhang, M.; Aminabadi, R.Y.; Awan, A.A.; Rasley, J.; He, Y. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. In Proceedings of the International conference on machine learning. PMLR, 2022, pp. 18332–18346.
9. Yao, J.; Anthony, Q.; Shafi, A.; Subramoni, H.; et al. Exploiting Inter-Layer Expert Affinity for Accelerating Mixture-of-Experts Model Inference. *arXiv preprint arXiv:2401.08383* **2024**.
10. Nie, X.; Miao, X.; Wang, Z.; Yang, Z.; Xue, J.; Ma, L.; Cao, G.; Cui, B. Flexmoe: Scaling large-scale sparse pre-trained model training via dynamic device placement. *Proceedings of the ACM on Management of Data* **2023**, *1*, 1–19.
11. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; Casas, D.d.l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv preprint arXiv:2310.06825* **2023**.
12. Muqeeth, M.; Liu, H.; Raffel, C. Soft merging of experts with adaptive routing. *arXiv preprint arXiv:2306.03745* **2023**.
13. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* **1992**, *8*, 229–256.
14. He, J.; Qiu, J.; Zeng, A.; Yang, Z.; Zhai, J.; Tang, J. Fastmoe: A fast mixture-of-expert training system. *arXiv preprint arXiv:2103.13262* **2021**.
15. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* **2023**.
16. Bi, X.; Chen, D.; Chen, G.; Chen, S.; Dai, D.; Deng, C.; Ding, H.; Dong, K.; Du, Q.; Fu, Z.; et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954* **2024**.
17. Jacobs, R.A.; Jordan, M.I.; Nowlan, S.J.; Hinton, G.E. Adaptive mixtures of local experts. *Neural computation* **1991**, *3*, 79–87.
18. Fedus, W.; Dean, J.; Zoph, B. A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667* **2022**.
19. Gou, Y.; Liu, Z.; Chen, K.; Hong, L.; Xu, H.; Li, A.; Yeung, D.Y.; Kwok, J.T.; Zhang, Y. Mixture of cluster-conditional lora experts for vision-language instruction tuning. *arXiv preprint arXiv:2312.12379* **2023**.
20. Puigcerver, J.; Ruiz, C.R.; Mustafa, B.; Houlsby, N. From Sparse to Soft Mixtures of Experts. In Proceedings of the The Twelfth International Conference on Learning Representations, 2023.
21. Zhang, X.; Shen, Y.; Huang, Z.; Zhou, J.; Rong, W.; Xiong, Z. Mixture of Attention Heads: Selecting Attention Heads Per Token. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 4150–4162.
22. Almahairi, A.; Ballas, N.; Cooijmans, T.; Zheng, Y.; Larochelle, H.; Courville, A. Dynamic capacity networks. In Proceedings of the International Conference on Machine Learning. PMLR, 2016, pp. 2549–2558.
23. Gao, C.; Chen, K.; Rao, J.; Sun, B.; Liu, R.; Peng, D.; Zhang, Y.; Guo, X.; Yang, J.; Subrahmanian, V. Higher Layers Need More LoRA Experts. *arXiv preprint arXiv:2402.08562* **2024**.
24. Li, Z.; You, C.; Bhojanapalli, S.; Li, D.; Rawat, A.S.; Reddi, S.J.; Ye, K.; Chern, F.; Yu, F.; Guo, R.; et al. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. *arXiv preprint arXiv:2210.06313* **2022**.
25. Mao, Y.; Mathias, L.; Hou, R.; Almahairi, A.; Ma, H.; Han, J.; Yih, S.; Khabsa, M. UniPELT: A Unified Framework for Parameter-Efficient Language Model Tuning. In Proceedings of the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 6253–6264.
26. Wang, H.; Polo, F.M.; Sun, Y.; Kundu, S.; Xing, E.; Yurochkin, M. Fusing Models with Complementary Expertise. In Proceedings of the The Twelfth International Conference on Learning Representations, 2023.
27. Chen, S.; Jie, Z.; Ma, L. Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms. *arXiv preprint arXiv:2401.16160* **2024**.
28. Kudugunta, S.; Huang, Y.; Bapna, A.; Krikun, M.; Lepikhin, D.; Luong, M.T.; Firat, O. Beyond Distillation: Task-level Mixture-of-Experts for Efficient Inference. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, 2021, pp. 3577–3599.

 doi:10.20944/preprints202508.0288.v1

29. Wang, Y.; Agarwal, S.; Mukherjee, S.; Liu, X.; Gao, J.; Awadallah, A.H.; Gao, J. AdaMix: Mixture-of-Adaptations for Parameter-efficient Model Tuning. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing; Goldberg, Y.; Kozareva, Z.; Zhang, Y., Eds., Abu Dhabi, United Arab Emirates, 2022; pp. 5744–5760. https://doi.org/10.18653/v1/2022.emnlp-main.388.
30. Ma, Z.; He, J.; Qiu, J.; Cao, H.; Wang, Y.; Sun, Z.; Zheng, L.; Wang, H.; Tang, S.; Zheng, T.; et al. BaGuaLu: targeting brain scale pretrained models with over 37 million cores. In Proceedings of the Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, 2022, pp. 192–204.
31. Komatsuzaki, A.; Puigcerver, J.; Lee-Thorp, J.; Ruiz, C.R.; Mustafa, B.; Ainslie, J.; Tay, Y.; Dehghani, M.; Houlsby, N. Sparse Upcycling: Training Mixture-of-Experts from Dense Checkpoints. In Proceedings of the The Eleventh International Conference on Learning Representations, 2022.
32. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research* **2023**, *24*, 1–113.
33. Kim, Y.J.; Awan, A.A.; Muzio, A.; Salinas, A.F.C.; Lu, L.; Hendy, A.; Rajbhandari, S.; He, Y.; Awadalla, H.H. Scalable and efficient moe training for multitask multilingual models. *arXiv preprint arXiv:2109.10465* **2021**.
34. Team, L.M. LLaMA-MoE: Building Mixture-of-Experts from LLaMA with Continual Pre-training, 2023.
35. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
36. Wu, X.; Huang, S.; Wei, F. MoLE: Mixture of LoRA Experts. In Proceedings of the The Twelfth International Conference on Learning Representations, 2023.
37. Chen, W.; Zhou, Y.; Du, N.; Huang, Y.; Laudon, J.; Chen, Z.; Cui, C. Lifelong language pretraining with distribution-specialized experts. In Proceedings of the International Conference on Machine Learning. PMLR, 2023, pp. 5383–5395.
38. Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; Chen, Z. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668* **2020**.
39. Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; Casas, D.d.L.; Hendricks, L.A.; Welbl, J.; Clark, A.; et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556* **2022**.
40. Rosenbaum, C.; Cases, I.; Riemer, M.; Klinger, T. Routing networks and the challenges of modular and compositional computation. *arXiv preprint arXiv:1904.12774* **2019**.
41. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. In Proceedings of the International Conference on Machine Learning. PMLR, 2019, pp. 2790–2799.
42. Han, Z.; Gao, C.; Liu, J.; Zhang, S.Q.; et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608* **2024**.
43. Zhang, Z.; Liu, S.; Yu, J.; Cai, Q.; Zhao, X.; Zhang, C.; Liu, Z.; Liu, Q.; Zhao, H.; Hu, L.; et al. M3oE: Multi-Domain Multi-Task Mixture-of Experts Recommendation Framework. In Proceedings of the Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 893–902.
44. Databricks. Introducing DBRX: A New State-of-the-Art Open LLM, 2024.
45. Clark, A.; de Las Casas, D.; Guy, A.; Mensch, A.; Paganini, M.; Hoffmann, J.; Damoc, B.; Hechtman, B.; Cai, T.; Borgeaud, S.; et al. Unified scaling laws for routed language models. In Proceedings of the International conference on machine learning. PMLR, 2022, pp. 4057–4086.
46. 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* **2023**.
47. Shen, Y.; Guo, Z.; Cai, T.; Qin, Z. JetMoE: Reaching Llama2 Performance with 0.1 M Dollars. *arXiv preprint arXiv:2404.07413* **2024**.
48. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10012–10022.
49. Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300* **2020**.

 doi:10.20944/preprints202508.0288.v1

50. Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.M.; Chen, W.; et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904* **2022**.
51. Tan, S.; Shen, Y.; Chen, Z.; Courville, A.; Gan, C. Sparse Universal Transformer. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 169–179.
52. Cai, W.; Jiang, J.; Qin, L.; Cui, J.; Kim, S.; Huang, J. Shortcut-connected Expert Parallelism for Accelerating Mixture-of-Experts. *arXiv preprint arXiv:2404.05019* **2024**.
53. Wei, T.; Zhao, L.; Zhang, L.; Zhu, B.; Wang, L.; Yang, H.; Li, B.; Cheng, C.; Lü, W.; Hu, R.; et al. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341* **2023**.
54. Shuster, K.; Xu, J.; Komeili, M.; Ju, D.; Smith, E.M.; Roller, S.; Ung, M.; Chen, M.; Arora, K.; Lane, J.; et al. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188* **2022**.
55. Wu, S.; Luo, J.; Chen, X.; Li, L.; Zhao, X.; Yu, T.; Wang, C.; Wang, Y.; Wang, F.; Qiao, W.; et al. Yuan 2.0-M32: Mixture of Experts with Attention Router. *arXiv preprint arXiv:2405.17976* **2024**.
56. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
57. Ren, J.; Rajbhandari, S.; Aminabadi, R.Y.; Ruwase, O.; Yang, S.; Zhang, M.; Li, D.; He, Y. {Zero-offload}: Democratizing {billion-scale} model training. In Proceedings of the 2021 USENIX Annual Technical Conference (USENIX ATC 21), 2021, pp. 551–564.
58. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D.; et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* **2022**, *35*, 24824–24837.
59. Shahbaba, B.; Neal, R. Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research* **2009**, *10*.
60. Xu, J.; Lai, J.; Huang, Y. MeteoRA: Multiple-tasks Embedded LoRA for Large Language Models. *arXiv preprint arXiv:2405.13053* **2024**.
61. Aghajanyan, A.; Gupta, S.; Zettlemoyer, L. Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 7319–7328.
62. Zheng, L.; Li, Z.; Zhang, H.; Zhuang, Y.; Chen, Z.; Huang, Y.; Wang, Y.; Xu, Y.; Zhuo, D.; Xing, E.P.; et al. Alpa: Automating inter-and {Intra-Operator} parallelism for distributed deep learning. In Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22), 2022, pp. 559–578.
63. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* **2016**.
64. Riquelme, C.; Puigcerver, J.; Mustafa, B.; Neumann, M.; Jenatton, R.; Susano Pinto, A.; Keysers, D.; Houlsby, N. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems* **2021**, *34*, 8583–8595.
65. Gross, S.; Ranzato, M.; Szlam, A. Hard mixtures of experts for large scale weakly supervised vision. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6865–6873.
66. Zhang, Z.; Lin, Y.; Liu, Z.; Li, P.; Sun, M.; Zhou, J. MoEfication: Transformer Feed-forward Layers are Mixtures of Experts. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, 2022, pp. 877–890.
67. Costa-jussà, M.R.; Cross, J.; Çelebi, O.; Elbayad, M.; Heafield, K.; Heffernan, K.; Kalbassi, E.; Lam, J.; Licht, D.; Maillard, J.; et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672* **2022**.
68. He, S.; Fan, R.Z.; Ding, L.; Shen, L.; Zhou, T.; Tao, D. Merging Experts into One: Improving Computational Efficiency of Mixture of Experts. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 14685–14691.
69. Lialin, V.; Deshpande, V.; Rumshisky, A. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647* **2023**.
70. Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874* **2021**.

 doi:10.20944/preprints202508.0288.v1

71. Chen, T.; Zhang, Z.; JAISWAL, A.K.; Liu, S.; Wang, Z. Sparse MoE as the New Dropout: Scaling Dense and Self-Slimmable Transformers. In Proceedings of the The Eleventh International Conference on Learning Representations, 2022.
72. Zhu, D.; Chen, J.; Shen, X.; Li, X.; Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* **2023**.
73. Roller, S.; Sukhbaatar, S.; Weston, J.; et al. Hash layers for large sparse models. *Advances in Neural Information Processing Systems* **2021**, *34*, 17555–17566.
74. Dou, S.; Zhou, E.; Liu, Y.; Gao, S.; Zhao, J.; Shen, W.; Zhou, Y.; Xi, Z.; Wang, X.; Fan, X.; et al. Loramoe: Revolutionizing mixture of experts for maintaining world knowledge in language model alignment. *arXiv preprint arXiv:2312.09979* **2023**.
75. Shen, Y.; Zhang, Z.; Cao, T.; Tan, S.; Chen, Z.; Gan, C. Moduleformer: Learning modular large language models from uncurated data. *arXiv preprint arXiv:2306.04640* **2023**.
76. Lin, B.; Tang, Z.; Ye, Y.; Cui, J.; Zhu, B.; Jin, P.; Zhang, J.; Ning, M.; Yuan, L. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947* **2024**.
77. Gao, Z.F.; Liu, P.; Zhao, W.X.; Lu, Z.Y.; Wen, J.R. Parameter-efficient mixture-of-experts architecture for pre-trained language models. *arXiv preprint arXiv:2203.01104* **2022**.
78. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Advances in neural information processing systems* **2020**, *33*, 1877–1901.
79. Wang, X.; Yu, F.; Dunlap, L.; Ma, Y.A.; Wang, R.; Mirhoseini, A.; Darrell, T.; Gonzalez, J.E. Deep mixture of experts via shallow embedding. In Proceedings of the Uncertainty in artificial intelligence. PMLR, 2020, pp. 552–562.
80. Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* **2024**, *568*, 127063.
81. Diao, S.; Xu, T.; Xu, R.; Wang, J.; Zhang, T. Mixture-of-Domain-Adapters: Decoupling and Injecting Domain Knowledge to Pre-trained Language Models' Memories. In Proceedings of the The 61st Annual Meeting Of The Association For Computational Linguistics, 2023.
82. Zhu, J.; Zhu, X.; Wang, W.; Wang, X.; Li, H.; Wang, X.; Dai, J. Uni-perceiver-moe: Learning sparse generalist models with conditional moes. *Advances in Neural Information Processing Systems* **2022**, *35*, 2664–2678.
83. Xue, F.; He, X.; Ren, X.; Lou, Y.; You, Y. One student knows all experts know: From sparse to dense. *arXiv preprint arXiv:2201.10890* **2022**.
84. Tang, H.; Liu, J.; Zhao, M.; Gong, X. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In Proceedings of the Proceedings of the 14th ACM Conference on Recommender Systems, 2020, pp. 269–278.
85. He, S.; Ding, L.; Dong, D.; Liu, B.; Yu, F.; Tao, D. PAD-Net: An Efficient Framework for Dynamic Networks. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 14354–14366.
86. Zuo, S.; Zhang, Q.; Liang, C.; He, P.; Zhao, T.; Chen, W. MoEBERT: from BERT to Mixture-of-Experts via Importance-Guided Adaptation. In Proceedings of the Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2022, pp. 1610–1623.
87. Chi, Z.; Dong, L.; Huang, S.; Dai, D.; Ma, S.; Patra, B.; Singhal, S.; Bajaj, P.; Song, X.; Mao, X.L.; et al. On the representation collapse of sparse mixture of experts. *Advances in Neural Information Processing Systems* **2022**, *35*, 34600–34613.
88. Dou, S.; Zhou, E.; Liu, Y.; Gao, S.; Zhao, J.; Shen, W.; Zhou, Y.; Xi, Z.; Wang, X.; Fan, X.; et al. The Art of Balancing: Revolutionizing Mixture of Experts for Maintaining World Knowledge in Language Model Alignment. *arXiv preprint arXiv:2312.09979* **2023**.
89. Team, Q. Introducing Qwen1.5, 2024.
90. Baltrušaitis, T.; Ahuja, C.; Morency, L.P. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence* **2018**, *41*, 423–443.

 doi:10.20944/preprints202508.0288.v1

91. Chen, Z.; Shen, Y.; Ding, M.; Chen, Z.; Zhao, H.; Learned-Miller, E.G.; Gan, C. Mod-squad: Designing mixtures of experts as modular multi-task learners. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 11828–11837.
92. Guo, Y.; Cheng, Z.; Tang, X.; Lin, T. Dynamic Mixture of Experts: An Auto-Tuning Approach for Efficient Transformer Models. *arXiv preprint arXiv:2405.14297* **2024**.
93. Shen, S.; Yao, Z.; Li, C.; Darrell, T.; Keutzer, K.; He, Y. Scaling vision-language models with sparse mixture of experts. *arXiv preprint arXiv:2303.07226* **2023**.
94. Du, Y.; Zhao, S.; Zhao, D.; Ma, M.; Chen, Y.; Huo, L.; Yang, Q.; Xu, D.; Qin, B. MoGU: A Framework for Enhancing Safety of Open-Sourced LLMs While Preserving Their Usability. *arXiv preprint arXiv:2405.14488* **2024**.
95. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* **2017**, *114*, 3521–3526.
96. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 4582–4597.
97. Team, Q. Qwen1.5-MoE: Matching 7B Model Performance with 1/3 Activated Parameters", 2024.
98. Jiang, C.; Tian, Y.; Jia, Z.; Zheng, S.; Wu, C.; Wang, Y. Lancet: Accelerating Mixture-of-Experts Training via Whole Graph Computation-Communication Overlapping. *arXiv preprint arXiv:2404.19429* **2024**.
99. McKinzie, B.; Gan, Z.; Fauconnier, J.P.; Dodge, S.; Zhang, B.; Dufter, P.; Shah, D.; Du, X.; Peng, F.; Weers, F.; et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611* **2024**.
100. Pan, B.; Shen, Y.; Liu, H.; Mishra, M.; Zhang, G.; Oliva, A.; Raffel, C.; Panda, R. Dense Training, Sparse Inference: Rethinking Training of Mixture-of-Experts Language Models. *arXiv preprint arXiv:2404.05567* **2024**.
101. Hu, E.J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the International Conference on Learning Representations, 2021.
102. Cai, R.; Muralidharan, S.; Heinrich, G.; Yin, H.; Wang, Z.; Kautz, J.; Molchanov, P. Flextron: Many-in-One Flexible Large Language Model. In Proceedings of the Forty-first International Conference on Machine Learning.
103. Shen, S.; Hou, L.; Zhou, Y.; Du, N.; Longpre, S.; Wei, J.; Chung, H.W.; Zoph, B.; Fedus, W.; Chen, X.; et al. Mixture-of-experts meets instruction tuning: A winning combination for large language models. *arXiv preprint arXiv:2305.14705* **2023**.
104. Zadouri, T.; Üstün, A.; Ahmadian, A.; Ermiş, B.; Locatelli, A.; Hooker, S. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. *arXiv preprint arXiv:2309.05444* **2023**.
105. Luo, T.; Lei, J.; Lei, F.; Liu, W.; He, S.; Zhao, J.; Liu, K. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models. *arXiv preprint arXiv:2402.12851* **2024**.
106. Chen, T.; Zhang, Z.; JAISWAL, A.K.; Liu, S.; Wang, Z. Sparse MoE as the New Dropout: Scaling Dense and Self-Slimmable Transformers. In Proceedings of the The Eleventh International Conference on Learning Representations, 2023.
107. Zheng, N.; Jiang, H.; Zhang, Q.; Han, Z.; Ma, L.; Yang, Y.; Yang, F.; Zhang, C.; Qiu, L.; Yang, M.; et al. Pit: Optimization of dynamic sparse deep learning models via permutation invariant transformation. In Proceedings of the Proceedings of the 29th Symposium on Operating Systems Principles, 2023, pp. 331–347.
108. Choi, J.Y.; Kim, J.; Park, J.H.; Mok, W.L.; Lee, S. SMoP: Towards Efficient and Effective Prompt Tuning with Sparse Mixture-of-Prompts. In Proceedings of the The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.
109. Zuo, S.; Liu, X.; Jiao, J.; Kim, Y.J.; Hassan, H.; Zhang, R.; Gao, J.; Zhao, T. Taming Sparsely Activated Transformer with Stochastic Experts. In Proceedings of the International Conference on Learning Representations, 2021.

 doi:10.20944/preprints202508.0288.v1

110. Ostapenko, O.; Caccia, L.; Su, Z.; Le Roux, N.; Charlin, L.; Sordoni, A. A Case Study of Instruction Tuning with Mixture of Parameter-Efficient Experts. In Proceedings of the NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following, 2023.
111. Shazeer, N.; Cheng, Y.; Parmar, N.; Tran, D.; Vaswani, A.; Koanantakool, P.; Hawkins, P.; Lee, H.; Hong, M.; Young, C.; et al. Mesh-tensorflow: Deep learning for supercomputers. *Advances in neural information processing systems* **2018**, *31*.