Review

Not peer-reviewed version

MoE at Scale: From Modular Design to Deployment in Large-Scale Machine Learning Systems

Aliyya Hadia , Tariq Afifa , Fawzi Gamal *

Posted Date: 16 April 2025

doi: 10.20944/preprints202504.1313.v1

Keywords: Mixture of Experts; sparse models; Neural Network Scaling; routing strategies; expert specialization; Modular Networks; large-scale machine learning; load balancing; conditional computation; foundation models; distributed training; interpretability; federated learning; continual learning; multimodal learning

 doi:10.20944/preprints202504.1313.v1

Article

MoE at Scale: From Modular Design to Deployment in Large-Scale Machine Learning Systems

Aliyya Hadia, Tariq Afifa 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 rapidly emerged as a foundational building block for scaling deep neural networks efficiently, enabling models with hundreds of billions of parameters to be trained and deployed with only a fraction of their total capacity active per input. By conditionally activating a sparse subset of expert modules, MoEs decouple model capacity from computation cost, offering an elegant and powerful framework for modular representation learning. This survey provides a comprehensive and systematic review of the MoE literature, spanning early formulations in ensemble learning and hierarchical mixtures to modern sparse MoEs powering large-scale language and vision models. We categorize MoE architectures along dimensions of gating mechanisms, expert sparsity, hierarchical composition, and cross-domain generalization. Further, we examine core algorithmic components such as routing strategies, load balancing, training dynamics, expert specialization, and infrastructure-aware deployment. We explore their applications across natural language processing, computer vision, speech, and multi-modal learning, and highlight their impact on foundation model development. Despite their success, MoEs raise open challenges in routing stability, interpretability, dynamic capacity allocation, and continual learning, which we discuss in depth alongside emerging research directions including federated MoEs, compositional generalization, and neuro-symbolic expert modules. We conclude by identifying trends that point toward MoEs as a central abstraction for building efficient, modular, and general-purpose AI systems. This survey serves as both a foundational reference and a forward-looking roadmap for researchers and practitioners seeking to understand and advance the state of Mixture of Experts.

Keywords: Mixture of Experts; sparse models; Neural Network Scaling; routing strategies; expert specialization; Modular Networks; large-scale machine learning; load balancing; conditional computation; foundation models; distributed training; interpretability; federated learning; continual learning; multi-modal learning

1. Introduction

In recent years, the field of machine learning has witnessed an unprecedented growth in the scale and complexity of models, driven primarily by the increased availability of large-scale datasets and the corresponding computational resources to process them. Within this landscape, the *Mixture of Experts* (MoE) paradigm has emerged as a powerful and scalable architectural design for both supervised and unsupervised learning tasks. First introduced in the early 1990s by Jacobs et al. [1], MoE models offer a compelling solution to the fundamental challenge of efficiently distributing computation and specialization across multiple subnetworks, or "experts", coordinated by a gating mechanism that dynamically selects or weighs their contributions. The core idea behind Mixture of Experts is to decompose a complex learning problem into a set of simpler, more tractable subproblems, each handled by a dedicated expert [2]. This modular decomposition not only enhances interpretability and fault tolerance but also facilitates scalability by enabling conditional computation: only a subset of experts are activated per input, significantly reducing the computational footprint during both training and inference. In contrast to monolithic deep neural networks that process all inputs uniformly, MoE

 doi:10.20944/preprints202504.1313.v1

architectures allow for a sparse activation regime, often governed by learned, data-dependent gating functions. These properties make MoE a natural fit for resource-constrained environments, large-scale multi-task learning, and settings that demand personalized or domain-adaptive behavior. The revival of interest in MoE architectures has been largely catalyzed by their successful deployment in modern deep learning systems, particularly in natural language processing (NLP) and computer vision. Recent instantiations such as the Switch Transformer [3], GShard [4], and Pathways [**?**] have demonstrated state-of-the-art performance on a variety of benchmarks while leveraging hundreds of billions of parameters [5]. Crucially, these architectures exhibit favorable scaling laws, wherein model capacity can be increased significantly without a proportional increase in computational cost. This decoupling of capacity from compute is achieved by routing inputs through a sparse subset of experts, thereby achieving both efficiency and expressivity. Consequently, MoE models have become a cornerstone of recent efforts to build general-purpose, foundation models capable of handling a wide range of modalities and tasks [6]. Despite their empirical success, Mixture of Experts models introduce a number of theoretical and practical challenges that remain open research frontiers. These include but are not limited to: (1) the stability and convergence of sparse training regimes, (2) the formulation of optimal routing and gating mechanisms, (3) issues of load balancing and expert specialization, (4) robustness to adversarial or out-of-distribution inputs, and (5) interpretability and fairness in expert decision-making [7]. Moreover, the increased architectural complexity of MoE systems raises important questions around debugging, reproducibility, and deployment in safety-critical applications [8]. This survey aims to provide a comprehensive and systematic review of the literature on Mixture of Experts, encompassing their historical foundations, theoretical principles, algorithmic advances, and diverse applications. In doing so, we seek to unify a fragmented body of work that spans multiple subfields, including ensemble learning, modular deep networks, meta-learning, and distributed systems. Our goal is to equip researchers and practitioners with a structured understanding of the MoE paradigm, highlight key challenges and opportunities, and identify promising directions for future work.

1.1. Scope and Contributions

The scope of this survey is intentionally broad, reflecting the wide applicability and evolution of MoE architectures over time. We begin by tracing the historical trajectory of MoE from its inception in statistical learning theory to its modern incarnations in large-scale deep learning systems. We then delve into the formal underpinnings of MoE, discussing both probabilistic and non-probabilistic formulations, including hard and soft gating mechanisms, conditional computation, and sparse activation. We further examine architectural variants, training algorithms, regularization techniques, and hardware considerations [9]. Key contributions of this survey include:

- A unified taxonomy of Mixture of Experts models, covering both classical and contemporary formulations.
- A detailed analysis of gating mechanisms, including recent developments in differentiable routing, reinforcement learning-based gating, and top-k selection [10].
- A comparative study of training paradigms, highlighting trade-offs between sparse and dense expert activation, load balancing, and convergence behavior [11].
- An overview of prominent applications across NLP, vision, recommendation systems, and multi-modal learning.
- A critical discussion of current limitations and future research directions, with a particular focus on interpretability, robustness, and deployment challenges.

1.2. Organization of the Survey

The remainder of this survey is organized as follows. Section 2 introduces the theoretical foundations of Mixture of Experts, including probabilistic formulations and early models. Section 3 categorizes and reviews key architectural designs, with a focus on deep learning-based MoE. Section 4 surveys gating and routing strategies, both deterministic and stochastic [12]. Section 5 examines training

 doi:10.20944/preprints202504.1313.v1

procedures, regularization methods, and optimization challenges [13]. Section 6 presents applications of MoE in various domains. Section **??** discusses open challenges and potential research directions. Finally, Section 8 concludes the survey with a summary of findings.

Notation

Throughout the paper, we denote input vectors by $\mathbf{x} \in \mathbb{R}^d$, expert outputs by $f_i(\mathbf{x})$, and gating outputs by $g_i(\mathbf{x})$, where $i \in \{1, \ldots, M\}$ indexes over the M experts. The final output of a Mixture of Experts model is given by:

$$f_{\text{MoE}}(\mathbf{x}) = \sum_{i=1}^{M} g_i(\mathbf{x}) f_i(\mathbf{x}), \tag{1}$$

where the gating function $g_i(\cdot)$ satisfies $\sum_i g_i(\mathbf{x}) = 1$ and $g_i(\mathbf{x}) \geq 0$.

2. Theoretical Foundations of Mixture of Experts

The Mixture of Experts (MoE) framework is rooted in the principle of modular function approximation, where the overall function to be learned is expressed as a composition or weighted combination of simpler subfunctions, each implemented by a distinct expert model. This paradigm is motivated by the observation that complex real-world tasks often exhibit underlying structure or heterogeneity that is best modeled via specialization—i.e., different subregions of the input space or different modes of the data distribution are better handled by different models. Formally, a Mixture of Experts model partitions the input space $\mathcal{X} \subset \mathbb{R}^d$ into a collection of overlapping or non-overlapping regions, and assigns to each region a specialized predictor, known as an expert. A gating network determines the degree to which each expert contributes to the final prediction [14]. This results in a form of piecewise or context-dependent modeling, wherein the effective model complexity is dynamically adapted based on the input [15].

2.1. Probabilistic Formulation

The original MoE model proposed by Jacobs et al. [1] is grounded in a probabilistic framework. Given an input $\mathbf{x}$, the output is modeled as a conditional mixture distribution over expert outputs:

$$p(y \mid \mathbf{x}) = \sum_{i=1}^{M} p(i \mid \mathbf{x})\, p(y \mid \mathbf{x}, i), \tag{2}$$

where $p(i \mid \mathbf{x})$ is the gating function representing the probability of selecting expert i, and $p(y \mid \mathbf{x}, i)$ is the likelihood of the output under expert i. Learning involves maximizing the log-likelihood of the data using techniques such as Expectation-Maximization (EM), or equivalently minimizing the negative log-likelihood loss:

$$\mathcal{L} = -\sum_{n=1}^{N} \log \sum_{i=1}^{M} p(i \mid \mathbf{x}_n)\, p(y_n \mid \mathbf{x}_n, i) [16]. \tag{3}$$

This probabilistic viewpoint allows MoE to be interpreted as a latent variable model, where the expert assignment is a hidden variable. In the hard-assignment variant, only the expert with the highest gate value is selected, leading to simplified inference and training.

2.2. Deterministic and Neural Formulation

Modern MoE models, particularly those used in deep learning, often eschew fully probabilistic interpretations in favor of deterministic or neural approximations. In this setting, the gating function is typically implemented as a neural network $g : \mathbb{R}^d \to \mathbb{R}^M$, often followed by a softmax or sparsemax activation to ensure normalization:

$$\mathbf{g}(\mathbf{x}) = \text{softmax}(W_g \mathbf{x} + b_g), \tag{4}$$

 doi:10.20944/preprints202504.1313.v1

with the output given by:

$$f_{\text{MoE}}(\mathbf{x}) = \sum_{i=1}^{M} g_i(\mathbf{x}) f_i(\mathbf{x}), \tag{5}$$

where $f_i(\mathbf{x})$ denotes the output of the i-th expert, typically another neural network. This formulation enables end-to-end training via backpropagation and seamlessly integrates with standard deep learning toolkits [17].

2.3. Sparse Gating and Conditional Computation

To improve efficiency and scalability, modern architectures often impose sparsity constraints on the gating function. This leads to *conditional computation*, where only a small subset (e.g., top-k) of experts is activated per input. The sparse gating function is typically defined as:

$$g_i(\mathbf{x}) = \begin{cases} \tilde{g}_i(\mathbf{x}) & \text{if } i \in \text{Top-}k(\tilde{\mathbf{g}}(\mathbf{x})) \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where $\tilde{\mathbf{g}}(\mathbf{x})$ are the pre-activation logits. This sparsity introduces challenges in gradient computation, necessitating techniques such as straight-through estimators, noisy gating [18], or auxiliary load balancing losses to encourage uniform expert utilization [19].

2.4. Function Approximation and Universal Representation

Mixture of Experts can also be interpreted through the lens of function approximation theory. Under suitable assumptions, it can be shown that an MoE with sufficiently expressive experts and a rich gating function can approximate any measurable function on a compact domain to arbitrary accuracy. This connects MoE to classical universal approximation results in neural networks and kernel methods. Specifically, let $\mathcal{F}_i$ denote the function class for expert i, and assume each $f_i \in \mathcal{F}_i$ can approximate a restricted function over some region of the input space. Then, the MoE model can approximate any function $f : \mathcal{X} \to \mathbb{R}$ as:

$$f(\mathbf{x}) \approx \sum_{i=1}^{M} g_i(\mathbf{x}) f_i(\mathbf{x}), \tag{7}$$

where $g_i(\mathbf{x})$ acts as a smooth partition of unity. This theoretical justification underscores the expressive power of MoE and motivates their use in multi-modal and multi-task settings.

2.5. Connections to Related Models

Mixture of Experts shares connections with a number of other machine learning paradigms, including:

- **Ensemble Learning:** Unlike traditional ensembles that average over fixed experts, MoE employs input-dependent gating.
- **Decision Trees:** MoE can be viewed as a soft generalization of decision trees, where routing decisions are probabilistic and differentiable.
- **Hierarchical Models:** Hierarchical MoEs stack multiple levels of expert selection, enabling deep modular hierarchies.
- **Meta-learning:** The gating function can be interpreted as a meta-learner that decides which expert(s) to deploy.

2.6. Challenges in Theoretical Analysis

Despite the intuitive appeal of MoE, formal analysis remains challenging [20]. Key difficulties include:

 doi:10.20944/preprints202504.1313.v1

- *Non-convex optimization:* Jointly learning experts and gating functions introduces highly non-convex loss landscapes.
- *Expert collapse:* Without careful regularization, experts may converge to similar behaviors, reducing diversity [21].
- *Load imbalance:* Sparse activation can lead to uneven expert utilization, causing inefficiencies in training.

These issues have inspired a range of algorithmic innovations, which we explore in subsequent sections.

3. Architectures and Design Patterns for Mixture of Experts

The architectural design of Mixture of Experts (MoE) models plays a central role in determining their scalability, expressivity, and efficiency [22]. From early shallow networks to modern deep and sparsely-activated transformer-based systems, MoE architectures have undergone significant evolution to meet the demands of large-scale learning. In this section, we present a systematic taxonomy of architectural variants, analyze their design principles, and highlight the trade-offs involved in different configurations.

3.1. Classical vs. Deep MoE Architectures

Early MoE models, such as those proposed by Jordan and Jacobs [1], were relatively shallow, typically comprising a single layer of experts and a softmax-based gating network. These models were effective for moderate-scale tasks such as speech recognition and time-series prediction. However, their shallow nature limited their representational power. In contrast, modern MoE architectures leverage deep neural networks for both the expert models and the gating function. Deep experts, such as convolutional neural networks (CNNs) or transformer blocks, provide high-capacity function approximation, while deep gating networks enable context-aware routing decisions. This allows for hierarchical abstraction and specialization across layers, facilitating complex tasks such as machine translation, image classification, and multi-modal reasoning [23].

3.2. Flat vs. Hierarchical MoE

3.2.1. Flat MoE

In the flat MoE design, all experts operate at the same hierarchical level and are selected independently for each input. The gating function computes a single routing decision, typically via top-k selection, to activate a sparse subset of experts:

$$f_{\text{MoE}}(\mathbf{x}) = \sum_{i \in \mathcal{A}_k(\mathbf{x})} g_i(\mathbf{x}) f_i(\mathbf{x}), \tag{8}$$

where $\mathcal{A}_k(\mathbf{x})$ denotes the top-k experts chosen by the gating network [24]. This approach is simple and scalable, but may underutilize the representational capacity of deep networks when applied at multiple layers [25].

3.2.2. Hierarchical MoE

Hierarchical Mixture of Experts (H-MoE) introduces multiple levels of gating and expert selection, enabling nested specializations. A higher-level gating network selects among groups of lower-level experts or even other MoE modules [26]. This leads to compositional representations and more efficient parameter usage:

$$f_{\text{H-MoE}}(\mathbf{x}) = \sum_{j=1}^{J} h_j(\mathbf{x}) \left(\sum_{i \in \mathcal{A}_k^j(\mathbf{x})} g_{i|j}(\mathbf{x}) f_{i|j}(\mathbf{x}) \right), \tag{9}$$

 doi:10.20944/preprints202504.1313.v1

where $h_j(\mathbf{x})$ selects a high-level group, and $g_{i|j}(\mathbf{x})$ selects experts within that group [27]. Hierarchical designs have been explored in multi-task learning [**?**], few-shot learning, and modular neural architectures.

3.3. Expert Types

Experts in MoE architectures can vary widely in complexity and structure depending on the task. Common expert types include:

- **Linear experts:** Lightweight and analytically tractable; useful for early-stage MoE research and toy problems [28].
- **Multi-layer perceptrons (MLPs):** Widely used in feed-forward MoE layers in modern transformer-based models.
- **Convolutional experts:** Suitable for image and video tasks; support spatial inductive biases [29].
- **Transformer blocks:** Used in large-scale NLP models (e.g., GShard [4], Switch Transformer [3]) for high-capacity sequence modeling.
- **Task-specific modules:** Experts designed for specific modalities or domains, such as visual reasoning, tabular data, or graph neural networks.

In practice, experts are often architecturally identical but differentiated through training, leading to emergent specialization [30].

3.4. Gating Configurations and Routing Granularity

A crucial design decision in MoE architectures is the *granularity* of routing decisions. Several patterns exist:

- **Token-level routing:** Each token in a sequence selects its own experts independently. Common in NLP models with large batch sizes and variable input lengths.
- **Sample-level routing:** A single gating decision is made for the entire input sample. Useful when inputs are short or semantically cohesive (e.g., images).
- **Feature-group routing:** Experts are specialized based on feature subspaces (e.g., visual vs textual features in multi-modal inputs).
- **Layer-wise MoE:** MoE layers are inserted at specific positions in a deep network, such as the feedforward layers in transformers.

Token-level and layer-wise routing are the most prevalent in recent deep MoE architectures [31].

3.5. Sparsity Mechanisms

To enable conditional computation and efficient scaling, sparsity is introduced in the gating outputs. Popular mechanisms include:

- **Top-k gating:** The gating function selects the k experts with highest scores for each input and zeros out others.
- **Noisy top-k:** Adds Gaussian noise to logits to encourage exploration during training [18] [32].
- **Sparsemax:** Uses a differentiable projection onto the simplex to achieve sparsity [**?**].
- **Routing via reinforcement learning:** Treats expert selection as a discrete decision problem.

Sparsity reduces memory and compute overhead, but complicates optimization due to non-differentiability and imbalanced load.

 doi:10.20944/preprints202504.1313.v1

3.6. Load Balancing and Expert Utilization

To prevent degenerate solutions where a small subset of experts dominates, modern MoE architectures employ auxiliary loss functions to encourage uniform expert usage [33]. One widely used loss is the load-balancing loss from Switch Transformer [3]:

$$\mathcal{L}_{\text{balance}} = M \sum_{i=1}^{M} \hat{p}_i \hat{g}_i, \tag{10}$$

where $\hat{p}_i$ is the average routing probability and $\hat{g}_i$ is the fraction of tokens routed to expert i. Minimizing this loss encourages uniform assignment while maintaining the expressivity of the gating network.

3.7. Scalability and Distributed Implementation

Modern MoE systems such as GShard and Pathways are designed to scale across multiple devices and nodes, enabling training with trillions of parameters. Key implementation strategies include:

- **Expert parallelism:** Experts are distributed across devices, with each device hosting a subset of experts.
- **Token dispatching:** Input tokens are routed to appropriate devices based on gating decisions.
- **All-to-all communication:** Used to efficiently dispatch tokens and gather outputs in distributed settings.
- **Caching and quantization:** Employed to reduce memory and bandwidth requirements.

These techniques are critical for achieving high throughput and low latency in large-scale MoE systems.

3.8. Design Trade-offs

MoE architecture design involves several trade-offs:

- *Expressivity vs [34]. efficiency:* Deeper or more specialized experts offer greater capacity but increase latency [35].
- *Modularity vs. coordination:* Modular experts facilitate specialization, but require coordination via gating [36].
- *Sparsity vs. smoothness:* Sparse activation improves efficiency but may lead to optimization instability [37].
- *Scalability vs. communication overhead:* Distributed execution enables scaling but incurs communication costs [38].

Effective design requires balancing these factors based on task requirements, computational budget, and deployment constraints [39].

4. Gating Mechanisms and Routing Strategies

At the core of every Mixture of Experts (MoE) model lies the gating mechanism — the component responsible for dynamically selecting which expert(s) are activated in response to a given input [40]. The gating function directly controls conditional computation, model sparsity, and expert specialization, making it critical for both efficiency and performance. In this section, we dissect the spectrum of gating designs, ranging from classical soft routing to modern sparse and stochastic approaches, and discuss the corresponding routing strategies that determine how input data is dispatched to experts.

4.1. Overview of Gating Functions

The gating function $g : \mathcal{X} \to \mathbb{R}^M$ maps input $\mathbf{x}$ to a score or probability distribution over M experts. The final output of the MoE layer is a weighted sum of expert outputs:

$$f_{\text{MoE}}(\mathbf{x}) = \sum_{i=1}^{M} g_i(\mathbf{x}) f_i(\mathbf{x}), \tag{11}$$

 doi:10.20944/preprints202504.1313.v1

where $g_i(\mathbf{x})$ is the routing score for expert i, and $f_i(\mathbf{x})$ is the corresponding expert's output. Gating functions can be broadly categorized into:

- **Soft routing (dense gating):** All experts contribute to the output with non-zero weights [41].
- **Sparse routing (hard gating):** Only a subset of experts are activated per input, typically chosen via top-k selection.
- **Stochastic routing:** Expert selection involves sampling from a learned or fixed probability distribution.
- **Learned discrete routing:** Expert assignments are treated as latent variables and learned via REINFORCE, Gumbel-softmax, or variational methods.

4.2. Softmax and Dense Gating

The most canonical gating mechanism uses a softmax activation over gating logits:

$$g_i(\mathbf{x}) = \frac{\exp(w_i^\top \mathbf{x})}{\sum_{j=1}^{M} \exp(w_j^\top \mathbf{x})}, \tag{12}$$

ensuring that all $g_i(\mathbf{x}) \in (0, 1)$ and sum to one [4]. This dense gating allows all experts to contribute, but scales linearly with the number of experts in both compute and memory. Moreover, dense activation hinders the efficiency gains of conditional computation.

4.3. Top-k Sparse Gating

To reduce computational overhead, sparse gating restricts the number of active experts per input to a small subset, often $k = 1$ or $k = 2$. The *top-k gating* mechanism retains only the largest k gating scores and zeroes out the rest:

$$g_i(\mathbf{x}) = \begin{cases} \text{softmax}_k(w^\top \mathbf{x})_i & \text{if } i \in \mathcal{A}_k(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases}, \tag{13}$$

where $\mathcal{A}_k(\mathbf{x})$ denotes the indices of the top-k logits. Sparse gating significantly improves efficiency and allows scaling to thousands of experts, as demonstrated in architectures like Switch Transformer [3].

4.4. Noisy Top-k Routing

To promote better expert utilization and avoid premature specialization, *noisy gating* introduces stochasticity by adding Gaussian noise to gating logits:

$$\tilde{w}_i = w_i^\top \mathbf{x} + \mathcal{N}(0, \sigma^2), \quad g_i(\mathbf{x}) = \text{top-}k(\tilde{w}_i). \tag{14}$$

This approach encourages exploration of different expert paths during training and smooths the optimization landscape. However, it also introduces variance in gradient estimation, necessitating techniques such as gradient clipping and load balancing losses [42].

4.5. Gumbel-Softmax and Differentiable Sampling

The Gumbel-softmax (or Concrete) distribution [**? ?**] enables approximate differentiable sampling from a categorical distribution. This is useful for modeling discrete expert choices while preserving end-to-end differentiability:

$$g_i(\mathbf{x}) = \frac{\exp((w_i^\top \mathbf{x} + G_i)/\tau)}{\sum_{j=1}^{M} \exp((w_j^\top \mathbf{x} + G_j)/\tau)}, \tag{15}$$

where $G_i \sim \text{Gumbel}(0, 1)$ and τ is a temperature parameter. As $\tau \to 0$, the distribution becomes one-hot. This technique bridges the gap between discrete gating and continuous optimization [43].

 doi:10.20944/preprints202504.1313.v1

4.6. REINFORCE-Based Routing

An alternative approach to gating uses policy gradient methods to learn discrete routing decisions [44]. The gating network is treated as a policy $\pi(i \mid \mathbf{x})$, and the model is trained using REINFORCE:

$$\nabla_\theta \mathbb{E}_{i \sim \pi(\cdot|\mathbf{x})}[R_i] = \mathbb{E}_i[R_i \nabla_\theta \log \pi(i \mid \mathbf{x})], \tag{16}$$

where R_i is a reward signal derived from downstream loss or expert performance. While flexible, REINFORCE suffers from high variance and slow convergence, and is thus less common in large-scale MoE systems [45].

4.7. Fixed and Data-Independent Routing

In some settings, fixed or manually defined gating schemes are used [46]. For instance, experts can be assigned based on predefined input attributes (e.g., modality, language, task ID), or via hash-based routing functions. While simple and efficient, such methods lack adaptability and can lead to suboptimal expert utilization.

4.8. Token-Level vs. Sample-Level Routing

Gating granularity affects how fine-grained expert decisions are [47]. The two most common types are:

- **Token-level gating:** Each token (or feature) in the input independently chooses its experts [48]. This is prevalent in sequence models like Transformers, where each token can exploit a different subset of expert knowledge.
- **Sample-level gating:** A single gating decision is made per input sample [49]. This reduces routing overhead but may limit expressivity.

Hybrid variants also exist, where groups of tokens (e.g., image patches or sentence segments) share a common gating decision.

4.9. Auxiliary Losses for Routing Stability

To mitigate issues like expert collapse and imbalance, auxiliary losses are added to the main training objective. A commonly used formulation is the load-balancing loss:

$$\mathcal{L}_{\text{balance}} = M \sum_{i=1}^{M} \hat{p}_i \hat{g}_i, \tag{17}$$

where $\hat{p}_i$ is the average routing probability and $\hat{g}_i$ is the fraction of inputs sent to expert i. Other regularizers include entropy penalties, KL divergence to uniform, and variance-based loss terms to enforce diversity in expert usage.

4.10. Routing in Multi-Modal and Multi-Task Settings

Advanced MoE systems extend routing to operate over heterogeneous input modalities (e.g., vision, language, audio) or multiple tasks. In such cases, the gating function can be conditioned on input type, task embeddings, or even output history, allowing for dynamic specialization across both inputs and outputs. Hierarchical gating is often employed to first route to modality-specific experts, followed by fine-grained expert selection within each domain [50].

4.11. Challenges in Routing Design

Despite significant progress, gating mechanisms present several open challenges:

- *Optimization instability:* Discrete and sparse routing leads to non-smooth loss surfaces [51].
- *Expert underutilization:* Without proper regularization, experts may remain idle or collapse to similar behaviors.

 doi:10.20944/preprints202504.1313.v1

- *Inference overhead:* Routing introduces computational and communication bottlenecks during deployment.
- *Routing bias:* Gating functions may overfit to training patterns, reducing generalization.

Overcoming these limitations is an active area of research, as it directly impacts the scalability and deployability of MoE systems.

5. Training Challenges and Optimization Techniques

Despite their appealing properties, Mixture of Experts (MoE) models are notoriously difficult to train effectively [52]. The conditional computation and sparsity that enable scalability also lead to a range of optimization challenges, including expert collapse, load imbalance, routing instability, gradient sparsity, and increased variance in updates [53]. Moreover, distributed and parallelized implementations introduce additional system-level constraints. In this section, we dissect these issues and survey the key strategies developed to address them in both academic and industrial-scale MoE deployments.

5.1. Expert Collapse and Load Imbalance

A common failure mode in MoE training is *expert collapse*, where only a few experts receive meaningful updates, while others remain idle or underutilized. This can occur due to biased gating decisions early in training or overly confident routing scores that lead to deterministic selection [54]. To counteract this, auxiliary loss functions are introduced to promote balanced usage across experts. A widely adopted regularization is the load-balancing loss [3,18]:

$$\mathcal{L}_{\text{balance}} = M \sum_{i=1}^{M} \hat{p}_i \hat{g}_i, \tag{18}$$

where $\hat{p}_i = \mathbb{E}_{\mathbf{x}}[g_i(\mathbf{x})]$ is the expected routing probability and $\hat{g}_i$ is the fraction of data assigned to expert i [55]. This encourages uniform assignment, particularly in sparse top-k routing settings. Other strategies include:

- **Entropy regularization:** Maximizing the entropy of the gating distribution to prevent early convergence to degenerate solutions [56].
- **KL divergence to uniform:** Penalizing divergence from uniform routing to spread load more evenly.
- **Temperature annealing:** Starting with high-temperature (soft) routing and gradually reducing it to allow smoother transitions in expert utilization.

5.2. Gradient Sparsity and Routing Discontinuities

Sparse routing leads to highly sparse gradients: only the active experts receive updates per input, and the gating network's gradients are limited to the top-k entries. This makes optimization brittle and susceptible to high variance. Solutions include:

- **Stochastic routing:** Adding noise to gating logits (e.g., Noisy Top-k) encourages broader exploration and smoother gradients.
- **Soft or continuous approximations:** Methods like Gumbel-softmax allow for differentiable approximations to discrete routing decisions.
- **Warm-up schedules:** Begin training with dense or soft routing and gradually increase sparsity to allow experts to develop meaningful representations [57].
- **Straight-through estimators (STE):** Use hard routing in the forward pass and a soft surrogate for the backward pass to stabilize learning [58].

 doi:10.20944/preprints202504.1313.v1

5.3. Training Instability and Routing Bias

The dynamic nature of expert selection introduces non-stationarity in the input distribution to each expert, especially when gating decisions change rapidly during training. This problem is exacerbated by poorly initialized gating networks or inconsistent token-expert assignments [59]. Best practices include:

- **Slow gating updates:** Apply lower learning rates to gating network parameters to reduce oscillations [60].
- **Batch-level balancing:** Normalize routing scores within mini-batches to reduce variance across training steps [61].
- **Frozen routing phases:** Temporarily freeze gating decisions to stabilize learning within expert subnetworks.
- **Expert normalization:** Use batch norm or layer norm within experts to mitigate shifts in input distributions [62].

5.4. Expert Capacity and Token Dropping

When too many tokens are routed to a given expert, especially in top-k gating, the system may exceed the expert's processing capacity. To handle this, capacity constraints are introduced:

$$C_i = \min(\lceil \frac{\text{tokens}}{M} \cdot \text{capacity factor} \rceil, C_{\max}), \tag{19}$$

and tokens beyond this threshold are dropped or routed to backup experts. Drop policies include:

- **Token dropping:** Ignore excess tokens and exclude them from the forward/backward pass [63].
- **Random backup routing:** Route excess tokens to randomly chosen available experts [64].
- **Re-ranking fallback:** Select the next highest scoring expert(s) that still have available capacity [65].

The choice of policy affects convergence speed and fairness among experts [66].

5.5. Parallelism and Distributed Training

MoE models typically operate in the regime of billions to trillions of parameters. Efficient distributed training is essential and introduces challenges in both communication and synchronization [67]. Solutions include:

- **Expert parallelism:** Experts are sharded across devices, enabling scalable computation with local memory constraints [68].
- **All-to-all token dispatching:** After routing, tokens must be sent to the appropriate device holding the selected experts.
- **Communication-efficient routing:** Use compact routing indices and fused communication kernels to reduce latency [69].
- **Gradient accumulation:** Accumulate gradients locally before global synchronization to minimize network traffic.

Frameworks such as GShard [4], DeepSpeed [**?**], and FairScale implement these strategies to enable efficient MoE training across thousands of GPUs or TPUs.

5.6. Regularization and Generalization

The modular nature of MoE models can lead to overfitting if individual experts memorize training patterns. To promote generalization, several regularization techniques are employed:

- **Dropout at expert level:** Randomly deactivate entire experts during training [70].
- **Expert dropout / switch dropout:** Randomly mask or shuffle expert assignments to encourage robustness.

 doi:10.20944/preprints202504.1313.v1

- **Weight sharing across experts:** Partially share parameters (e.g., input/output layers) to reduce overparameterization.
- **Inter-expert consistency losses:** Penalize disagreement among experts for similar inputs to enforce coherent behavior.

These strategies enhance sample efficiency and reduce reliance on any single expert.

5.7. Curriculum Learning and Progressive Sparsification

Some training regimens adopt a curriculum learning approach where sparsity is introduced gradually [71]. For instance:

- Start with dense routing and low capacity thresholds [72].
- Gradually increase sparsity via temperature annealing or increasing the top-k constraint [73].
- Freeze expert parameters periodically to promote routing stability.

This helps avoid early collapse, improves load distribution, and yields smoother convergence.

5.8. Monitoring and Debugging MoE Training

Due to the dynamic and modular nature of MoEs, new tooling is often required to monitor training progress:

- **Expert utilization histograms:** Visualize token assignments across experts.
- **Routing entropy metrics:** Quantify the diversity and sharpness of gating outputs [74].
- **Token routing traces:** Track token flow through experts over time [75].
- **Gradient heatmaps:** Identify sparsity and imbalance in expert updates.

These diagnostics are essential for understanding failures and tuning routing parameters [76].

5.9. Summary of Training Strategies

Table 1 summarizes the main training challenges and corresponding mitigation techniques [77].

Table 1. Summary of MoE training challenges and mitigation strategies.

Challenge	Mitigation Strategies
Expert collapse	Load-balancing loss, entropy regularization, capacity limits
Sparse gradients	Gumbel-softmax, stochastic routing, STE, warm-up
Routing instability	Learning rate schedules, frozen routing, batch normalization
Overloaded experts	Token dropping, backup routing, re-ranking fallback
Communication bottlenecks	All-to-all dispatching, gradient accumulation, fused kernels
Overfitting	Expert dropout, weight sharing, inter-expert consistency losses

6. Applications of Mixture of Experts

Mixture of Experts (MoE) models have demonstrated state-of-the-art performance across a broad spectrum of machine learning domains, particularly where scale, diversity of input, and task complexity demand modularity and conditional computation. Their ability to scale model capacity without linearly increasing computational cost makes them well-suited for large-scale settings such as natural language processing, computer vision, speech processing, and multi-modal learning. In this section, we review the principal applications of MoEs, highlight the unique advantages they offer in each domain, and outline domain-specific adaptations and deployment considerations.

6.1. Natural Language Processing (NLP)

MoE models have seen the most traction in natural language processing, especially in the context of large language models (LLMs) [78]. The modular nature of MoEs fits naturally with the token-based

 doi:10.20944/preprints202504.1313.v1

structure of text, allowing individual tokens or token groups to be routed to different experts based on semantic or syntactic features.

Language Modeling and Pretraining.

Pioneering models such as Switch Transformer [3], GShard [4], and GLaM [79] have demonstrated how sparse MoE architectures can scale to hundreds of billions of parameters while maintaining constant computational cost per token. These models integrate MoE layers into Transformer blocks, where routing decisions are made per token and a small number of experts (e.g., top-k) are activated per forward pass [80].

Multilingual Models.

MoEs have proven particularly effective in multilingual settings, where they facilitate dynamic specialization across languages and scripts [81]. For instance, GShard uses language-specific experts with load-balancing constraints to improve translation quality while sharing a single model backbone[50,82]. Similar ideas appear in M6-T [**?**] and M2M-100 [83], which leverage expert specialization to reduce interference between low-resource and high-resource languages.

Fine-tuning and Adaptation.

MoEs also support parameter-efficient fine-tuning by enabling task-specific experts or adapter-style experts that are activated only for certain domains. This improves data efficiency and allows large pretrained MoE models to be adapted to downstream tasks with minimal retraining [84].

6.2. Computer Vision

Although vision applications traditionally favor dense computation due to spatial correlations, MoE-style architectures have recently gained ground through hybrid and sparse attention mechanisms.

Vision Transformers (ViTs) with MoE.

Works such as Vision MoE [85] integrate sparse expert layers into Vision Transformers, improving performance on classification and segmentation tasks while maintaining throughput. Routing can be applied at the patch level, enabling dynamic token-expert assignments based on visual features [86].

Multi-Scale and Region-Based Routing.

In detection and segmentation tasks, experts can specialize in regions of interest (RoIs) or scales of input (e.g., coarse vs. fine-grained features). Conditional convolution and deformable attention have been extended into expert frameworks to achieve this dynamic specialization [87].

Challenges in Vision.

Compared to NLP, vision-based MoEs face additional challenges such as alignment between image patches and expert capacity, increased routing overhead, and sensitivity to noise in visual inputs. Efforts to address these include hierarchical MoE architectures and region-of-interest gating strategies.

6.3. Speech and Audio Processing

In speech recognition and synthesis, MoE models offer the ability to specialize across phonetic structures, accents, speakers, and modalities (e.g., waveform, spectrogram, linguistic units) [88].

Speaker and Dialect Specialization.

Sparse expert layers have been used to improve robustness across diverse speaker identities and languages by allowing experts to specialize in phonetic patterns. This leads to improvements in recognition accuracy, particularly in low-resource and noisy environments.

 doi:10.20944/preprints202504.1313.v1

Conditional Generative Models.

In text-to-speech and audio generation, expert specialization can be applied to control attributes such as pitch, timbre, and emotion. MoEs enable these factors to be modeled independently while sharing a common generative backbone.

6.4. Multi-Modal Learning

MoEs are naturally suited to multi-modal architectures, where different data modalities (text, image, audio, video) require distinct inductive biases and feature transformations [89].

Modality-Specific Experts.

In multi-modal Transformers such as Perceiver IO [**?**] and Flamingo [**?**], experts can be assigned based on input modality (e.g., text vs. image) or combined features, allowing the model to dynamically adapt its processing pipeline.

Cross-Modal Routing.

Some models use learned gating conditioned on both modalities to route inputs to fusion experts or joint representation layers. This allows for complex interactions and hierarchical fusion across modalities [90].

Alignment and Synchronization.

One challenge in multi-modal MoEs is ensuring temporal or spatial alignment across modalities. Techniques such as synchronized gating and modality-aware top-k selection help maintain coherence [91].

6.5. Personalization and Recommendation Systems

In large-scale recommendation systems, user preferences, histories, and item categories vary widely, making MoEs ideal for modeling diverse behaviors.

User/Item-Specific Experts.

Personalized MoEs route user-item pairs to specialized submodels, improving ranking accuracy and CTR (click-through rate) [92]. Experts may be learned via user clustering, hashed from IDs, or adaptively trained.

Scalability and Latency.

Deploying MoEs in recommendation pipelines requires careful engineering to meet latency constraints[93]. Lightweight experts and routing approximations (e.g., hashing, locality-sensitive gating) are often used to optimize inference throughput [94].

6.6. Reinforcement Learning and Robotics

In reinforcement learning (RL), MoEs have been used to modularize policy learning, state abstraction, and value estimation across different environments or skills.

Policy Composition.

MoEs allow for skill-based decomposition where each expert represents a distinct behavior or subtask [95]. Gating is conditioned on state, goal, or environment context, supporting transfer and lifelong learning [96].

Exploration and Diversity.

By routing similar states to different experts over time, MoEs can enhance exploration and prevent policy collapse in sparse-reward settings. This has been explored in mixture policy gradients and options-based learning.

 doi:10.20944/preprints202504.1313.v1

6.7. Emerging Use Cases and Research Directions

Beyond established domains, MoEs are beginning to influence a range of new applications:

- **Federated learning:** Experts are trained locally on decentralized data, and gating coordinates knowledge sharing across clients.
- **Continual learning:** MoEs offer a framework for allocating new experts to novel tasks without catastrophic forgetting [97].
- **Neurosymbolic systems:** Experts implement differentiable modules for logic, planning, or rule-based inference.

6.8. Summary and Deployment Considerations

Table 2 summarizes key application domains of MoEs, associated benefits, and deployment constraints.

Table 2. Application domains of Mixture of Experts and key considerations.

Domain	MoE Advantages	Challenges and Constraints
NLP	Scalability, language specialization	Load balancing, routing overhead
Vision	Token-wise routing, region specialization	Patch alignment, noise sensitivity
Speech	Speaker adaptability, temporal gating	Capacity planning, modality shifts
Multi-modal	Modality-specific routing, fusion flexibility	Synchronization, expert sharing
Personalization	User-tailored modeling, CTR boost	Latency, memory footprint
RL/Robotics	Skill modularity, transfer learning	Non-stationarity, exploration bias

7. Open Challenges and Future Directions

While Mixture of Experts (MoE) architectures have made remarkable strides in scaling neural networks efficiently, numerous theoretical and practical challenges remain unresolved. As MoEs are deployed in increasingly diverse and critical applications—from large language models to embodied agents—the need to better understand, interpret, and refine their behavior grows correspondingly urgent. In this section, we survey the principal open challenges and chart promising directions for future research [98].

7.1. Scalability Limits and Inference Trade-offs

MoEs promise sublinear scaling of computational cost with model capacity [99]. However, practical limits exist due to:

- **Communication overhead:** All-to-all dispatching between devices becomes a bottleneck at extreme scales [100].
- **Expert duplication:** Memory footprint grows linearly with the number of experts unless sharing mechanisms are introduced [101].
- **Batch fragmentation:** Sparse routing can lead to uneven mini-batch sizes per expert, impacting throughput and utilization [102].
- **Latency variance:** Routing-dependent compute paths introduce variability in inference latency, which is problematic in real-time systems [103].

 doi:10.20944/preprints202504.1313.v1

Future Directions.

Research is ongoing into token coalescing, sparse attention-aware graph routing, and expert caching mechanisms to reduce communication cost [104]. There is also interest in hybrid architectures that blend dense and sparse computation adaptively [105].

7.2. Generalization vs. Specialization Trade-off

A central tension in MoEs is between specialization (experts learning distinct capabilities) and generalization (broad applicability across inputs) [106]. Highly specialized experts can overfit or fragment representation space, while overly general experts dilute the benefits of modularity.

Research Questions.

- What is the optimal number of experts per input distribution?
- How does expert specialization impact robustness to distribution shift?
- Can we learn interpretable expert functions without explicit supervision [107]?

Directions Forward.

Emerging work in meta-learning and information bottleneck theory offers tools to quantify and guide specialization [108]. Cross-expert consistency losses and auxiliary classifiers are also used to balance this trade-off.

7.3. Routing Interpretability and Trustworthiness

MoE models rely heavily on learned routing decisions. However, these decisions are often opaque and sensitive to perturbations, leading to concerns about model robustness and explainability.

Key Challenges.

- **Unstable routing:** Small input changes may lead to large changes in routing paths [109].
- **Opaque logic:** Gating networks may encode heuristics that are difficult to inspect [110].
- **Security risks:** Malicious routing patterns can be exploited to bias model outputs.

Possible Solutions.

Interpretable routing via attention-based gating, logic-based templates, or symbolic gating modules could offer better transparency. Visualizations of token flow and expert attention maps are another useful tool.

7.4. Sparse Training and Inference Synergy

Training-time sparsity does not always align with inference-time constraints [111]. For example, sparsity benefits in pretraining may not translate to low-latency inference due to routing costs and hardware inefficiencies [112].

Research Goals.

- Align sparse training dynamics with efficient inference pipelines [113].
- Develop sparsity-aware hardware primitives and compilers.
- Train MoEs with explicit cost-aware objectives (e.g., latency, FLOPs) [114].

Methods such as reinforcement learning-based routing policies or hardware-in-the-loop training are being explored to bridge this gap [115].

7.5. Foundation Models and Modular Scaling

Foundation models (FMs), such as GPT, PaLM, and Flamingo, are increasingly being equipped with MoE layers to expand capacity and specialization [116]. However, modular scaling introduces its own set of challenges:

 doi:10.20944/preprints202504.1313.v1

- **Expert reuse across tasks:** How should experts be reused, adapted, or frozen across tasks [117]?
- **Backward compatibility:** Can expert modules be updated independently without retraining the full model?
- **Composable modularity:** Can experts be composed dynamically at runtime based on user intent or context [118]?

Frameworks such as modular transformers and compositional neural architectures are promising directions to tackle these questions.

7.6. Continual, Lifelong, and Federated Learning

MoEs offer a natural framework for continual learning, where new experts can be allocated to new tasks, and catastrophic forgetting is mitigated by parameter isolation.

Open Problems.

- **Expert proliferation:** How can we limit unbounded expert growth over time?
- **Dynamic capacity allocation:** Can the model adaptively allocate compute to evolving distributions [119]?
- **Federated MoEs:** Can experts be learned locally across devices and aggregated without centralization [120]?

Lifelong MoEs, elastic routing, and expert pruning/growth strategies are active research areas. In federated settings, MoEs enable partial sharing of global and local knowledge, reducing bandwidth costs and preserving privacy.

7.7. Neurosymbolic and Compositional Reasoning

There is increasing interest in using MoEs to implement compositional or symbolic reasoning in neural models [121]. For example, different experts might encode mathematical operations, logical templates, or procedural rules [122].

Challenges and Questions.

- Can routing networks learn to compose experts to simulate algorithmic pipelines [123]?
- How do we supervise or regularize symbolic behavior in experts?
- What are the limits of compositionality achievable with MoEs [124]?

Integrations with program induction, neural module networks, and logic-based supervision are promising directions for future neuro-symbolic MoEs.

7.8. Unified Evaluation Benchmarks

Despite widespread adoption, there is a lack of standardized evaluation benchmarks for MoE architectures across domains [125]. Metrics such as utilization, routing entropy, robustness, and interpretability are often ignored in favor of task-specific accuracy [126].

Proposal.

A unified MoE benchmark suite should include:

- Diverse tasks across NLP, vision, speech, and RL.
- Routing diagnostics and expert efficiency metrics [127].
- Stress tests for robustness, sparsity, and capacity overload.

Such a benchmark would facilitate more principled comparisons across different MoE designs [128].

7.9. Summary

Table 3 summarizes key open challenges and corresponding research opportunities [1].

 doi:10.20944/preprints202504.1313.v1

Table 3. Open challenges in Mixture of Experts and potential research directions.

Challenge	Future Research Directions
Scalability limits	Expert caching, sparse communication graphs, memory sharing
Specialization vs. generalization	Meta-routing, expert consistency regularization
Interpretability	Symbolic gating, routing visualization tools
Training-inference mismatch	Cost-aware training, RL-based routing policies
Modular scaling for FMs	Compositional experts, backward-compatible updates
Continual/federated learning	Dynamic expert allocation, local/global parameter isolation
Neuro-symbolic MoEs	Programmatic experts, differentiable interpreters
Benchmarking	Unified metrics, routing entropy, utilization variance

8. Conclusion

Mixture of Experts (MoE) architectures have emerged as a powerful paradigm for enabling scalable, modular, and efficient neural computation. By decoupling capacity from compute via conditional execution, MoEs offer a compelling solution to the increasing demands of large-scale machine learning, particularly in the context of foundation models across language, vision, speech, and multi-modal domains.

This survey has provided a comprehensive examination of the MoE landscape, beginning with a historical overview of early developments and continuing through recent advances in sparse routing, dynamic gating, and expert regularization. We have classified and analyzed a broad family of MoE architectures, highlighting their diverse formulations—from hard versus soft gating, to top-k routing, capacity-constrained balancing, and hierarchical expert selection. Our treatment has emphasized not only architectural innovations, but also practical training techniques, initialization strategies, load-balancing constraints, and routing algorithms that address the unique optimization dynamics of sparse models.

A key theme throughout has been the growing relevance of MoEs in real-world applications. In natural language processing, MoEs underpin the most efficient large-scale language models to date, allowing for orders of magnitude greater capacity without linear increases in compute. In computer vision and speech, MoEs enable task-adaptive and region-specific modeling, pushing the boundaries of representational flexibility. In multi-modal systems, MoEs naturally facilitate modular fusion of heterogeneous inputs, unlocking new capabilities in embodied reasoning, perception, and communication.

Despite this progress, numerous challenges remain unresolved. MoEs introduce complex trade-offs between specialization and generalization, raise concerns around interpretability and routing stability, and expose bottlenecks in distributed infrastructure. Furthermore, the absence of unified benchmarks and theoretical frameworks limits the ability to compare or rigorously understand different MoE strategies. These limitations signal not the decline but the nascency of a paradigm—one whose full potential has yet to be realized.

Looking ahead, MoEs are poised to play a central role in the next generation of intelligent systems. Their modularity offers a pathway to lifelong learning, continual adaptation, and reusable components in artificial intelligence. As the community begins to explore compositional reasoning, neuro-symbolic integration, and personalized AI at scale, MoEs provide the architectural scaffolding for such modular cognition.

Ultimately, the trajectory of MoEs will depend on continued innovation across multiple fronts: algorithmic, theoretical, and infrastructural. Advances in hardware-aware routing, interpretability tools, and hybrid dense-sparse models may soon make MoEs the default choice for both pretraining and deployment of large models. As researchers and practitioners converge on this frontier, the

 doi:10.20944/preprints202504.1313.v1

Mixture of Experts framework will likely become a foundational abstraction in the machine learning toolkit—a powerful blend of capacity, efficiency, and modular intelligence.

References

1. Jacobs, R.A.; Jordan, M.I.; Nowlan, S.J.; Hinton, G.E. Adaptive mixtures of local experts. *Neural computation* **1991**, *3*, 79–87.
2. Shahbaba, B.; Neal, R. Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research* **2009**, *10*.
3. 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.
4. 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**.
5. 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**.
6. 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.
7. 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. *Advances in Neural Information Processing Systems* **2022**, *35*, 7103–7114.
8. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.
9. 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.
10. Xu, J.; Lai, J.; Huang, Y. MeteoRA: Multiple-tasks Embedded LoRA for Large Language Models. *arXiv preprint arXiv:2405.13053* **2024**.
11. 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**.
12. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* **2020**.
13. Shazeer, N. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202* **2020**.
14. Singh, S.; Ruwase, O.; Awan, A.A.; Rajbhandari, S.; He, Y.; Bhatele, A. A Hybrid Tensor-Expert-Data Parallelism Approach to Optimize Mixture-of-Experts Training. In Proceedings of the Proceedings of the 37th International Conference on Supercomputing, 2023, pp. 203–214.
15. 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.
16. Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. Efficient large scale language modeling with mixtures of experts. *arXiv preprint arXiv:2112.10684*, 2021
17. Team, Q. Introducing Qwen1.5, 2024.
18. Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* **2017**.
19. 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.
20. Wu, J.; Hu, X.; Wang, Y.; Pang, B.; Soricut, R. Omni-SMoLA: Boosting Generalist Multimodal Models with Soft Mixture of Low-rank Experts. *arXiv preprint arXiv:2312.00968* **2023**.
21. 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*.
22. 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**.

 doi:10.20944/preprints202504.1313.v1

23. 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.
24. 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**.
25. 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.
26. Zhang, B.; Sennrich, R. Root mean square layer normalization. *Advances in Neural Information Processing Systems* **2019**, *32*.
27. 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.
28. Lialin, V.; Deshpande, V.; Rumshisky, A. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647* **2023**.
29. 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**.
30. 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**.
31. Huang, C.; Liu, Q.; Lin, B.Y.; Pang, T.; Du, C.; Lin, M. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269* **2023**.
32. 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.
33. Liu, H.; Tam, D.; Muqeeth, M.; Mohta, J.; Huang, T.; Bansal, M.; Raffel, C.A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems* **2022**, *35*, 1950–1965.
34. Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; Chi, E.H. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In Proceedings of the Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 1930–1939.
35. Rajbhandari, S.; Rasley, J.; Ruwase, O.; He, Y. Zero: Memory optimizations toward training trillion parameter models. In Proceedings of the SC20: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2020, pp. 1–16.
36. Theis, L.; Bethge, M. Generative image modeling using spatial lstms. *Advances in neural information processing systems* **2015**, *28*.
37. Zheng, Y.; Wang, D.X. A survey of recommender systems with multi-objective optimization. *Neurocomputing* **2022**, *474*, 141–153.
38. Rosenbaum, C.; Klinger, T.; Riemer, M. Routing networks: Adaptive selection of non-linear functions for multi-task learning. *arXiv preprint arXiv:1711.01239* **2017**.
39. Li, Y.; Hui, B.; Yin, Z.; Yang, M.; Huang, F.; Li, Y. PaCE: Unified Multi-modal Dialogue Pre-training with Progressive and Compositional Experts. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 13402–13416.
40. 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.
41. 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.
42. 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**.
43. Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C.H.; Gonzalez, J.; Zhang, H.; Stoica, I. Efficient memory management for large language model serving with pagedattention. In Proceedings of the Proceedings of the 29th Symposium on Operating Systems Principles, 2023, pp. 611–626.

 doi:10.20944/preprints202504.1313.v1

44. 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**.
45. Shoeybi, M.; Patwary, M.; Puri, R.; LeGresley, P.; Casper, J.; Catanzaro, B. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053* **2019**.
46. 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.
47. Hwang, C.; Cui, W.; Xiong, Y.; Yang, Z.; Liu, Z.; Hu, H.; Wang, Z.; Salas, R.; Jose, J.; Ram, P.; et al. Tutel: Adaptive mixture-of-experts at scale. *Proceedings of Machine Learning and Systems* **2023**, *5*.
48. 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**.
49. 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**.
50. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
51. 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.
52. Narayanan, D.; Shoeybi, M.; Casper, J.; LeGresley, P.; Patwary, M.; Korthikanti, V.; Vainbrand, D.; Kashinkunti, P.; Bernauer, J.; Catanzaro, B.; et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In Proceedings of the Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2021, pp. 1–15.
53. 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**.
54. 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.
55. 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.
56. Zhou, K.; Yang, J.; Loy, C.C.; Liu, Z. Learning to prompt for vision-language models. *International Journal of Computer Vision* **2022**, *130*, 2337–2348.
57. 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.
58. Mustafa, B.; Riquelme, C.; Puigcerver, J.; Jenatton, R.; Houlsby, N. Multimodal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems* **2022**, *35*, 9564–9576.
59. 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**.
60. 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.
61. Muqeeth, M.; Liu, H.; Raffel, C. Soft merging of experts with adaptive routing. *arXiv preprint arXiv:2306.03745* **2023**.
62. 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**.
63. 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**.
64. Zhu, Y.; Wichers, N.; Lin, C.C.; Wang, X.; Chen, T.; Shu, L.; Lu, H.; Liu, C.; Luo, L.; Chen, J.; et al. Sira: Sparse mixture of low rank adaptation. *arXiv preprint arXiv:2311.09179* **2023**.

 doi:10.20944/preprints202504.1313.v1

65. Nie, X.; Miao, X.; Cao, S.; Ma, L.; Liu, Q.; Xue, J.; Miao, Y.; Liu, Y.; Yang, Z.; Cui, B. Evomoe: An evolutional mixture-of-experts training framework via dense-to-sparse gate. *arXiv preprint arXiv:2112.14397* **2021**.
66. Gu, A.; Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752* **2023**.
67. Liu, Q.; Wu, X.; Zhao, X.; Zhu, Y.; Xu, D.; Tian, F.; Zheng, Y. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *arXiv preprint arXiv:2310.18339* **2023**.
68. 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.
69. 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**.
70. Shen, L.; Wu, Z.; Gong, W.; Hao, H.; Bai, Y.; Wu, H.; Wu, X.; Bian, J.; Xiong, H.; Yu, D.; et al. Se-moe: A scalable and efficient mixture-of-experts distributed training and inference system. *arXiv preprint arXiv:2205.10034* **2022**.
71. 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**.
72. Dat, D.H.; Mao, P.Y.; Nguyen, T.H.; Buntine, W.; Bennamoun, M. HOMOE: A Memory-Based and Composition-Aware Framework for Zero-Shot Learning with Hopfield Network and Soft Mixture of Experts. *arXiv preprint arXiv:2311.14747* **2023**.
73. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
74. Tan, S.; Shen, Y.; Panda, R.; Courville, A. Scattered Mixture-of-Experts Implementation. *arXiv preprint arXiv:2403.08245* **2024**.
75. 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**.
76. Li, D.; Ma, Y.; Wang, N.; Cheng, Z.; Duan, L.; Zuo, J.; Yang, C.; Tang, M. MixLoRA: Enhancing Large Language Models Fine-Tuning with LoRA based Mixture of Experts. *arXiv preprint arXiv:2404.15159* **2024**.
77. 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.
78. 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**.
79. Du, N.; Huang, Y.; Dai, A.M.; Tong, S.; Lepikhin, D.; Xu, Y.; Krikun, M.; Zhou, Y.; Yu, A.W.; Firat, O.; et al. Glam: Efficient scaling of language models with mixture-of-experts. In Proceedings of the International Conference on Machine Learning. PMLR, 2022, pp. 5547–5569.
80. Chen, G.; Liu, F.; Meng, Z.; Liang, S. Revisiting Parameter-Efficient Tuning: Are We Really There Yet? 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. 2612–2626. https://doi.org/10.18653/v1/2022.emnlp-main.168.
81. OpenAI. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt, 2022.
82. Bahng, H.; Jahanian, A.; Sankaranarayanan, S.; Isola, P. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274* **2022**.
83. Fan, A.; Bhosale, S.; Schwenk, H.; Ma, Z.; El-Kishky, A.; Goyal, S.; Baines, M.; Celebi, O.; Wenzek, G.; Chaudhary, V.; et al. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research* **2021**, *22*, 1–48.
84. 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**.
85. 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.
86. Zhang, P.; Li, X.; Hu, X.; Yang, J.; Zhang, L.; Wang, L.; Choi, Y.; Gao, J. Vinvl: Revisiting visual representations in vision-language models. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 5579–5588.

 doi:10.20944/preprints202504.1313.v1

87. 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.
88. Zhang, Z.; Zeng, Z.; Lin, Y.; Xiao, C.; Wang, X.; Han, X.; Liu, Z.; Xie, R.; Sun, M.; Zhou, J. Emergent Modularity in Pre-trained Transformers. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 4066–4083.
89. Jiang, J.; Wang, F.; Shen, J.; Kim, S.; Kim, S. A Survey on Large Language Models for Code Generation. *arXiv preprint arXiv:2406.00515* **2024**.
90. Zhao, H.; Qiu, Z.; Wu, H.; Wang, Z.; He, Z.; Fu, J. HyperMoE: Towards Better Mixture of Experts via Transferring Among Experts. *arXiv preprint arXiv:2402.12656* **2024**.
91. 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.
92. Qi, P.; Wan, X.; Huang, G.; Lin, M. Zero Bubble Pipeline Parallelism. In Proceedings of the The Twelfth International Conference on Learning Representations, 2023.
93. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
94. Li, M.; Gururangan, S.; Dettmers, T.; Lewis, M.; Althoff, T.; Smith, N.A.; Zettlemoyer, L. Branch-Train-Merge: Embarrassingly Parallel Training of Expert Language Models. In Proceedings of the First Workshop on Interpolation Regularizers and Beyond at NeurIPS 2022, 2022.
95. Yang, A.; Lin, J.; Men, R.; Zhou, C.; Jiang, L.; Jia, X.; Wang, A.; Zhang, J.; Wang, J.; Li, Y.; et al. M6-t: Exploring sparse expert models and beyond. *arXiv preprint arXiv:2105.15082* **2021**.
96. Sukhbaatar, S.; Golovneva, O.; Sharma, V.; Xu, H.; Lin, X.V.; Rozière, B.; Kahn, J.; Li, D.; Yih, W.t.; Weston, J.; et al. Branch-Train-MiX: Mixing Expert LLMs into a Mixture-of-Experts LLM. *arXiv preprint arXiv:2403.07816* **2024**.
97. Yoo, K.M.; Han, J.; In, S.; Jeon, H.; Jeong, J.; Kang, J.; Kim, H.; Kim, K.M.; Kim, M.; Kim, S.; et al. HyperCLOVA X Technical Report. *arXiv preprint arXiv:2404.01954* **2024**.
98. 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**.
99. Shi, S.; Pan, X.; Chu, X.; Li, B. PipeMoE: Accelerating Mixture-of-Experts through Adaptive Pipelining. In Proceedings of the IEEE INFOCOM 2023-IEEE Conference on Computer Communications. IEEE, 2023, pp. 1–10.
100. Lu, J.; Batra, D.; Parikh, D.; Lee, S. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems* **2019**, *32*.
101. Zhong, Z.; Xia, M.; Chen, D.; Lewis, M. Lory: Fully Differentiable Mixture-of-Experts for Autoregressive Language Model Pre-training. *arXiv preprint arXiv:2405.03133* **2024**.
102. 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**.
103. 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.
104. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* **1992**, *8*, 229–256.
105. 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.
106. He, S.; Dong, D.; Ding, L.; Li, A. Demystifying the Compression of Mixture-of-Experts Through a Unified Framework. *arXiv preprint arXiv:2406.02500* **2024**.
107. Huang, Y.; Cheng, Y.; Bapna, A.; Firat, O.; Chen, D.; Chen, M.; Lee, H.; Ngiam, J.; Le, Q.V.; Wu, Y.; et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems* **2019**, *32*.
108. 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.
109. Liu, Y.; Zhang, R.; Yang, H.; Keutzer, K.; Du, Y.; Du, L.; Zhang, S. Intuition-aware Mixture-of-Rank-1-Experts for Parameter Efficient Finetuning. *arXiv preprint arXiv:2404.08985* **2024**.

 doi:10.20944/preprints202504.1313.v1

110. Zhang, Z.; Xia, Y.; Wang, H.; Yang, D.; Hu, C.; Zhou, X.; Cheng, D. MPMoE: Memory Efficient MoE for Pre-trained Models with Adaptive Pipeline Parallelism. *IEEE Transactions on Parallel and Distributed Systems* **2024**.
111. Nie, X.; Zhao, P.; Miao, X.; Zhao, T.; Cui, B. HetuMoE: An efficient trillion-scale mixture-of-expert distributed training system. *arXiv preprint arXiv:2203.14685* **2022**.
112. 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.
113. 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**.
114. Raposo, D.; Ritter, S.; Richards, B.; Lillicrap, T.; Humphreys, P.C.; Santoro, A. Mixture-of-Depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258* **2024**.
115. Wei, J.; Bosma, M.; Zhao, V.; Guu, K.; Yu, A.W.; Lester, B.; Du, N.; Dai, A.M.; Le, Q.V. Finetuned Language Models are Zero-Shot Learners. In Proceedings of the International Conference on Learning Representations, 2021.
116. 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.
117. Shen, Y.; Guo, Z.; Cai, T.; Qin, Z. JetMoE: Reaching Llama2 Performance with 0.1 M Dollars. *arXiv preprint arXiv:2404.07413* **2024**.
118. DeepSeek-AI. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model, 2024, [arXiv:cs.CL/2405.04434].
119. Ma, X.; Zhao, L.; Huang, G.; Wang, Z.; Hu, Z.; Zhu, X.; Gai, K. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In Proceedings of the The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 1137–1140.
120. Wu, X.; Huang, S.; Wei, F. Mixture of LoRA Experts. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
121. Gururangan, S.; Lewis, M.; Holtzman, A.; Smith, N.A.; Zettlemoyer, L. DEMix Layers: Disentangling Domains for Modular Language Modeling. 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. 5557–5576.
122. 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.
123. Li, S.; Xue, F.; Baranwal, C.; Li, Y.; You, Y. Sequence parallelism: Long sequence training from system perspective. *arXiv preprint arXiv:2105.13120* **2021**.
124. Roller, S.; Sukhbaatar, S.; Weston, J.; et al. Hash layers for large sparse models. *Advances in Neural Information Processing Systems* **2021**, *34*, 17555–17566.
125. Bengio, Y.; Léonard, N.; Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* **2013**.
126. 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.
127. Shi, C.; Yang, C.; Zhu, X.; Wang, J.; Wu, T.; Li, S.; Cai, D.; Yang, Y.; Meng, Y. Unchosen Experts Can Contribute Too: Unleashing MoE Models' Power by Self-Contrast. *arXiv preprint arXiv:2405.14507* **2024**.
128. Uppal, S.; Bhagat, S.; Hazarika, D.; Majumder, N.; Poria, S.; Zimmermann, R.; Zadeh, A. Multimodal research in vision and language: A review of current and emerging trends. *Information Fusion* **2022**, *77*, 149–171.