Route, Select, Activate: The Mechanics of Mixture of Experts

Fawzi Gamal[1]

[1]Department of Computer Science and Technology,

King Abdullah University of Science and Technology

Email: fawzi.gamal@kaust.edu.sa

Abstract

Mixture of Experts (MoE) has emerged as a foundational architectural principle in deep learning, enabling the construction of models that are both highly expressive and computationally efficient. By activating only a sparse subset of specialized expert networks for each input, MoE architectures effectively decouple model capacity from compute cost, offering a scalable alternative to traditional monolithic models. This paradigm has seen a resurgence in recent years, with breakthroughs in large-scale language modeling, vision, speech, and multi-modal learning, culminating in some of the most powerful models to date, such as Switch Transformer, GLaM, and V-MoE. This survey presents a comprehensive and systematic overview of Mixture of Experts in the context of deep learning. We begin with a historical perspective on the origins of MoE, tracing its evolution from classical ensemble methods to modern, sparsely-activated architectures. The core design choices of MoE systems—including expert sparsity, gating functions, routing strategies, and hierarchical extensions—are carefully dissected. We examine a wide array of training challenges, such as expert imbalance, routing instability, and optimization bottlenecks, along with proposed solutions including load-balancing regularizers, top-k routing approximations, expert dropout, and gradient routing relaxation. A central focus of this survey is the application landscape of MoE, spanning natural language processing, computer vision, speech and audio processing, multi-modal tasks, and continual learning. Across these domains, MoE has demonstrated compelling gains in scalability, generalization, and modular adaptability. In parallel, we explore the theoretical underpinnings of MoE, connecting it to concepts from ensemble learning, conditional computation, modularity, and universal function approximation. We also discuss emerging insights into MoE's

generalization behavior, routing dynamics, and optimization landscape. Despite its promise, MoE remains an active and evolving field. We conclude by identifying open challenges—ranging from routing robustness and interpretability to dynamic expert generation and deployment constraints—and outlining directions for future research. In synthesizing current knowledge and charting future paths, this survey aims to serve as a definitive resource for both researchers and practitioners seeking to understand and harness the potential of Mixture of Experts in deep learning.

Keywords: Mixture of Experts, Conditional Computation, Sparse Deep Models, Expert Routing, Scalable Architectures.

1 Introduction

In recent years, the field of deep learning has undergone a dramatic transformation, giving rise to models of unprecedented scale and complexity that have revolutionized a wide array of tasks in computer vision, natural language processing, speech recognition, and beyond. Amidst this rapid evolution, one paradigm that has garnered increasing attention is the *Mixture of Experts* (MoE) framework. Initially proposed in the early 1990s as a form of ensemble learning rooted in the divide-and-conquer principle, MoE has experienced a resurgence in the deep learning era, emerging as a powerful strategy for achieving scalability, modularity, and computational efficiency in large-scale neural networks [1]. The core idea behind Mixture of Experts is deceptively simple yet profoundly effective: rather than relying on a single monolithic model to learn all aspects of a task, an MoE architecture leverages a collection of specialized sub-models, referred to as "experts," each of which is responsible for learning a subset of the problem space. A gating mechanism—often a neural network in its own right—determines which experts are most relevant for a given input and selectively activates a sparse subset of them during inference [2]. This conditional computation paradigm enables MoE models to allocate computational resources dynamically, allowing for significant reductions in inference cost while maintaining or even improving performance [3]. In the deep learning context, Mixture of Experts architectures are uniquely positioned to address several of the most pressing challenges facing modern neural networks. Chief among these is the tension between scale and efficiency [4]. As models continue to grow—exemplified by large language models with hundreds of billions of parameters—the computational and environmental costs of training and inference become increasingly prohibitive. MoE models offer a compelling solution by enabling sparse activation: only a fraction of the model is used per input, thereby decoupling model capacity from computational overhead. This property has led to the development of some of the largest and most efficient models to

date, including sparsely activated transformers such as the Switch Transformer and GLaM. Another compelling advantage of the MoE framework lies in its inherent modularity and interpretability [5]. The division of labor among experts allows for clearer attribution of function, making it easier to analyze, debug, and refine individual components of the model. Moreover, this modularity supports continual learning and transfer learning paradigms by enabling targeted updates to specific experts without affecting the entire system. As such, MoE models are not only scalable but also adaptable and extensible, attributes that are crucial for the development of robust and sustainable artificial intelligence systems [6]. Despite its advantages, the practical realization of MoE models introduces a host of technical challenges [7]. These include, but are not limited to, the design of effective gating mechanisms, the prevention of expert collapse (where a small subset of experts dominate computation), the efficient routing of inputs in distributed environments, and the stability of training dynamics. A variety of strategies have been proposed to address these issues, ranging from reinforcement learning-based gates and entropy regularization, to auxiliary load-balancing losses and top-k routing heuristics [8]. Furthermore, recent innovations have extended the MoE paradigm to novel settings, including unsupervised learning, multimodal processing, and federated learning, highlighting the versatility and generality of the approach [9]. The aim of this survey is to provide a comprehensive and critical overview of Mixture of Experts in the context of deep learning [10]. We trace the historical development of MoE models, elucidate their theoretical foundations, and examine the diverse ways in which they have been instantiated in modern architectures. We also explore the breadth of applications that have benefited from MoE-based models, from natural language understanding and image classification, to machine translation and recommendation systems. Particular attention is given to recent advances that have pushed the boundaries of what is possible with MoE, as well as to open challenges that remain unresolved [11]. We organize the rest of this paper as follows [12]. In Section 2, we provide the necessary background on the classical MoE framework and its evolution into deep learning contexts. Section 4 surveys key architectural innovations in MoE, including gating strategies, expert design, and routing techniques. Section 5 discusses the training algorithms, optimization challenges, and solutions that have been proposed. In Section 6, we present a taxonomy of applications that have successfully leveraged MoE models. Section **??** offers an in-depth analysis of the empirical performance and theoretical properties of MoE. Finally, in Section **??**, we outline emerging trends, ongoing research efforts, and promising directions for future work in this rapidly evolving field [13]. Through this survey, we aim to distill the fundamental insights and practical lessons learned from the extensive body of work on Mixture of Experts, and to serve as a guide for researchers and practitioners seeking to harness the power

of this paradigm in their own domains. As the deep learning community continues to grapple with the challenges of scale, efficiency, and generalization, we believe that MoE will play an increasingly central role in shaping the next generation of intelligent systems.

2 Background

The concept of Mixture of Experts (MoE) originates from the early studies in ensemble learning and modular neural networks in the 1990s [14]. Initially introduced by Jacobs et al. [15], the foundational idea behind MoE is to decompose complex learning tasks into smaller, more manageable sub-tasks, each of which is handled by an individual expert. The original formulation combined multiple neural networks (the experts), each trained to specialize in a subset of the input space, with a gating network that learned to assign probabilistic weights to each expert's output based on the input instance. Formally, given an input $\mathbf{x}$, a traditional MoE model computes the output as a weighted sum of the experts' predictions:

$$y(\mathbf{x}) = \sum_{i=1}^{K} g_i(\mathbf{x}) E_i(\mathbf{x}), \tag{1}$$

where $E_i(\mathbf{x})$ is the output of the i-th expert and $g_i(\mathbf{x})$ is the gating function's output for expert i, satisfying $\sum_i g_i(\mathbf{x}) = 1$. This formulation allows each expert to focus on specific regions of the input distribution while the gating function orchestrates the collaboration among them [16]. The appeal of this framework lies in its alignment with the principles of specialization and conditional computation [17]. Instead of training a single model to master an entire task domain, MoE enables localized learning and inference, which can be significantly more efficient and interpretable [18]. Early successes of this method in smaller-scale settings, such as handwriting recognition and regression problems, provided strong empirical evidence for its effectiveness [19]. However, the complexity of training MoE systems—particularly the instability of the gating network and difficulties with expert utilization balance—posed significant obstacles to broader adoption at the time [20]. With the advent of deep learning and the availability of large-scale data and computational resources, the interest in MoE models has been reignited [21]. A pivotal moment came with the introduction of sparsely-gated MoE layers in large-scale models such as the Mixture of Experts layer in the Google Translate model [22]. In this formulation, only a small subset of experts is activated for each input (typically the top-k scoring experts according to the gating function), allowing the model to scale to billions of parameters while keeping per-example computational costs manageable [23]. This sparsity introduces challenges such as load balancing and expert under-utilization, but it also enables efficient training

across distributed systems, making MoE a cornerstone in the design of scalable transformer-based architectures. Modern MoE approaches can be broadly categorized into two families: **dense MoE**, where all experts contribute to the final output (as in the classical formulation), and **sparse MoE**, where a small number of experts are conditionally selected for each input [24]. Sparse MoEs have gained more traction in recent years due to their favorable trade-offs between computational cost and model capacity [25]. Notable implementations include Switch Transformers [26], which simplify the gating mechanism by selecting only one expert per input token, and GLaM [27], which enhances the routing mechanism with improved regularization and training stability [28]. Beyond the architectural design, MoE models are deeply intertwined with broader trends in deep learning [29]. The desire to scale models without proportional increases in compute has made conditional computation frameworks increasingly attractive [30]. Similarly, the pursuit of modular, reusable, and interpretable neural components aligns closely with the principles underpinning MoE. Furthermore, recent research has expanded the application of MoE beyond language modeling into vision [31], speech [**?**], and multi-modal settings [**?**]. To support these applications, MoE has also evolved in terms of training techniques. Challenges such as expert collapse (where a small subset of experts dominate the workload), training instability, and inefficient routing have spurred innovations in gating functions, auxiliary losses, expert assignment heuristics, and gradient balancing mechanisms [32]. For instance, the use of load-balancing losses [22], top-k routing [33], and capacity-aware gating strategies [33] are now standard practice in modern MoE implementations. In this survey, we adopt a comprehensive view of the MoE paradigm, encompassing both its classical roots and modern instantiations in deep neural architectures [34]. We aim to highlight the continuity and evolution of ideas, from early ensemble methods to today's large-scale, distributed models. In the next section, we delve into the architectural aspects of Mixture of Experts, categorizing and comparing the diverse implementations that have emerged over time.

3 Theoretical Foundations

Despite the practical success of Mixture of Experts (MoE) models across numerous domains, a rigorous theoretical understanding of their behavior remains an active area of research [35]. In this section, we examine the theoretical motivations behind MoE architectures, explore connections to ensemble learning, conditional computation, and modularity, and discuss recent efforts to formalize their generalization capacity, sample efficiency, and expressive power.

3.1 MoE as Conditional Computation

MoE models embody the principle of *conditional computation*—only a subset of the model's parameters are activated for any given input. This differs from conventional deep learning models, where all parameters are typically engaged per input [36]. From an information-theoretic perspective, conditional computation allows for greater representational capacity per unit of computation [37]. Specifically, given E experts and k-way routing, the number of effective model configurations grows combinatorially:

$$\text{Number of active sub-networks} = \binom{E}{k}, \tag{2}$$

which implies that sparse MoE models can approximate a wide variety of functions with fewer active parameters per input, provided that routing is effective.

3.2 Connections to Ensemble Learning

MoE architectures are closely related to ensemble learning methods, such as bagging, boosting, and stacking [38]. Classical ensemble theory suggests that aggregating diverse models can reduce generalization error, especially when base learners are weak and uncorrelated [39]. In MoE, each expert can be viewed as a specialized learner, and the gating function plays the role of a meta-learner that adaptively weights or selects among them [40]. Unlike traditional ensembles, however, MoE models benefit from joint training of experts and the gating network [41]. This introduces rich dynamics: experts may compete for routing probability, and the gating function evolves to balance both task performance and computational efficiency [26]. These interactions make MoE both more powerful and more complex than static ensembles.

3.3 Modularity and Specialization

One of the core theoretical appeals of MoE models lies in their ability to support modular representations. Modular architectures have long been hypothesized to improve sample efficiency, transferability, and interpretability. MoE models induce a soft partitioning of the input space, with each expert specializing in a subregion [42]. This leads to localized learning, where each expert only needs to model a subset of the input distribution. Recent work has investigated the benefits of such specialization from a learning-theoretic lens [43]. For instance, modular learning systems have been shown to reduce catastrophic interference by isolating gradient updates [44]. Furthermore, theoretical results from multi-task learning suggest that parameter decoupling (as in MoE) can improve generalization when task-relatedness is low or task boundaries are unknown.

3.4 Function Approximation and Expressive Power

The expressive power of MoE networks has been analyzed using universal approximation theory [45]. It has been shown that a sufficiently large MoE model with learned routing can approximate any piecewise smooth function arbitrarily well [**?**]. In particular, the use of piecewise linear experts (e.g., ReLU networks) combined with input-dependent gating introduces non-trivial compositionality and decision boundaries [46]. This approximation capability is further enhanced by hierarchical MoE structures, where routing occurs at multiple levels [47]. Such compositions allow for adaptive depth and width across the input space, improving both representation efficiency and sample complexity in high-dimensional regimes [48].

3.5 Generalization and Sample Efficiency

While overparameterized models typically risk overfitting, MoE architectures often generalize well despite their massive parameter counts [31]. This paradox has prompted theoretical inquiry into how conditional sparsity affects generalization bounds. Recent PAC-Bayesian analyses and Rademacher complexity bounds have been extended to sparse MoE models, indicating that the generalization error depends more on the number of active parameters per input than on total model size [49]. This aligns with empirical findings that MoE models with billions of parameters can outperform smaller dense models using the same computational budget [50]. Moreover, because different experts are trained on distinct subsets of data (as determined by routing), MoE models can benefit from implicit data clustering [51]. This can lead to improved sample efficiency, particularly when the data distribution exhibits latent structure or multimodality [52].

3.6 Routing Dynamics and Optimization Landscape

The routing function in MoE introduces a non-stationary target for expert networks [49]. As routing distributions evolve during training, the effective data seen by each expert changes [53]. This can lead to instability or convergence to suboptimal equilibria [54]. From an optimization perspective, MoE models are non-convex and piecewise differentiable, with discontinuities introduced by top-k selection. Nevertheless, certain variants—such as those using soft or continuous gating—enable smoother gradients and improved convergence properties [55]. Gradient-based game-theoretic frameworks have also been proposed to model the interaction between experts and gating as a multi-agent optimization problem [**?**]. Understanding the training dynamics of MoE is an open challenge, with ongoing work analyzing the stability of expert assignment, the effect of gating noise, and the emergence of specialization under different training regimes.

3.7 Theoretical Trade-offs

The design of MoE models involves balancing several theoretical trade-offs:

- **Capacity vs. Generalization**: Activating fewer experts improves regularization but may limit expressiveness [56].
- **Specialization vs. Redundancy**: Strong specialization enhances modularity but reduces robustness to routing errors.
- **Dynamic Routing vs. Optimization Stability**: Hard selection enables sharp decisions but complicates gradient flow [57].

These trade-offs highlight the need for principled strategies that can control the complexity of expert interactions while maintaining robustness and learning efficiency.

3.8 Summary

Theoretical insights into Mixture of Experts architectures reveal their rich connections to foundational principles in machine learning, including ensemble theory, modularity, and conditional computation [58]. While much progress has been made, key questions remain about the limits of generalization, the structure of the optimization landscape, and the dynamics of specialization. These insights are crucial for informing the design of next-generation MoE systems. In the following section, we shift our focus to limitations, challenges, and open problems, providing a roadmap for future research [59].

4 Architectural Design of Mixture of Experts

The architectural design of Mixture of Experts (MoE) models plays a central role in their effectiveness, scalability, and deployment efficiency. In this section, we provide a systematic overview of the key architectural components of MoE systems, including the expert networks, gating mechanisms, routing strategies, and integration into broader model topologies [60]. We further categorize prominent variants and extensions, discussing their trade-offs and practical implications.

4.1 Core Components

A typical MoE architecture comprises three primary components: a set of expert networks, a gating mechanism, and a routing function [61].

Expert Networks. Each expert in an MoE model is usually instantiated as a feed-forward neural network or a module compatible with the overarching model architecture. In transformer-based designs, for instance, experts are often positioned within the feed-forward layers (FFN), replacing the standard MLP block with an MoE layer composed of multiple parallel MLPs. The degree of diversity among experts varies across implementations—some models allow experts to share parameters partially or fully, while others enforce specialization through data partitioning or regularization.

Gating Mechanisms. The gating network is responsible for determining which experts should be activated for a given input [62]. In its simplest form, the gating function outputs a softmax distribution over the experts:

$$\mathbf{g}(\mathbf{x}) = \text{softmax}(W_g\mathbf{x} + \mathbf{b}_g), \tag{3}$$

where W_g and $\mathbf{b}_g$ are learnable parameters [63]. However, for sparse MoE models, this distribution is often truncated via top-k selection, and only the most relevant experts are activated [64]. The gating network may be trained jointly with the experts, with additional auxiliary objectives to promote balanced expert utilization [65].

Routing Functions. Routing determines how inputs are assigned to experts during training and inference. In dense MoE, each input is broadcast to all experts and combined via a weighted sum [66]. In contrast, sparse MoE uses a routing function that selects the top-k experts per input (often $k = 1$ or $k = 2$), significantly reducing computation [67]. Efficient routing requires careful design to avoid overloading certain experts and under-utilizing others. Popular routing strategies include Noisy Top-k, used in GShard [33], and Switch routing, where only one expert is activated per token [26] [68].

4.2 Dense vs [69]. Sparse MoE Architectures

Dense MoE architectures, while conceptually straightforward, suffer from scalability limitations due to the need to compute all expert outputs per input. These models tend to be more stable during training but are less efficient for large-scale deployments [70]. Sparse MoEs, by contrast, scale more gracefully by leveraging conditional computation [71]. However, their complexity introduces challenges such as gradient sparsity, load imbalance, and expert collapse. Sparse MoE models are typically preferred in practice due to their computational advantages. In these designs, each token or input is routed to a limited number of experts. For instance, GShard activates two experts per token and includes a capacity constraint to limit

token allocation per expert [72]. Switch Transformer goes further by selecting a single expert per token, significantly simplifying implementation and improving throughput on TPU-based systems [73].

4.3 Integration into Transformer Architectures

One of the most successful applications of MoE has been in the context of transformers, where MoE layers replace standard feed-forward sub-layers. This design enables model scaling without linearly increasing computation [74]. Notable transformer-MoE hybrids include:

- **GShard** [33]: Introduced sparsely-gated MoE layers with top-2 expert routing and auxiliary load balancing losses.
- **Switch Transformer** [26]: Simplified routing by using top-1 selection, achieving scalability with minimal degradation in performance.
- **GLaM** [27]: Utilized a mixture of dense and sparse experts to improve robustness and reduce overfitting.
- **V-MoE** [31]: Applied MoE to the vision domain, integrating sparse expert modules into vision transformer architectures.

These architectures demonstrate that MoE layers can serve as plug-and-play modules within larger deep learning frameworks, offering flexibility and extensibility.

4.4 Hierarchical and Multi-Level MoE

Recent work has also explored hierarchical MoE structures, where experts themselves contain internal routing mechanisms or are organized into tiers. This design allows for deeper modularization and can support tasks with complex structure or hierarchical dependencies [75]. Multi-level MoE systems have been applied to multi-task learning and hierarchical representation learning, where different levels of abstraction benefit from different degrees of specialization [76].

4.5 Parameter Sharing and Conditional Sparsity

While traditional MoE architectures assume distinct expert modules, some designs introduce partial parameter sharing to reduce memory footprint and enhance generalization. Shared-bottom MoE architectures, for example, allow experts to share lower-layer parameters while specializing at higher layers. Conditional sparsity

further extends this concept by dynamically adapting which parameters are used based on input properties, merging MoE ideas with neural pruning and dynamic networks [77].

4.6 Regularization and Load Balancing

To ensure that all experts are effectively utilized, modern MoE models incorporate regularization terms that penalize unbalanced routing. A common strategy is to introduce a load-balancing loss that encourages uniform usage across experts. For example, the auxiliary loss used in GShard minimizes the KL divergence between the actual expert usage distribution and a uniform distribution [78]. Other techniques include entropy regularization of gating outputs and noise injection during training to promote exploration [79].

4.7 Design Trade-offs and Practical Considerations

Designing an MoE architecture involves trade-offs across several axes:

- **Sparsity vs. Accuracy**: Increasing sparsity improves efficiency but may lead to lower model capacity utilization [80].
- **Routing Complexity vs [81]. Training Stability**: Complex routing improves specialization but can cause instability during training.
- **Expert Independence vs. Parameter Sharing**: Independent experts allow for specialization but increase memory usage.
- **Deployment Simplicity vs. Model Expressiveness**: Simple gating strategies ease deployment but may sacrifice nuanced expert combinations [82].

These trade-offs must be carefully balanced according to task requirements, hardware constraints, and training budget [83]. In the next section, we examine how these architectural components interact with training algorithms and optimization dynamics, exploring the unique challenges and solutions involved in training Mixture of Experts models effectively.

5 Training and Optimization

Training Mixture of Experts (MoE) models introduces a set of challenges that are significantly more complex than those found in standard deep learning architectures [84]. These challenges arise from the model's conditional computation

paradigm, the dynamic nature of expert routing, and the increased sparsity in gradient propagation. In this section, we explore the unique optimization dynamics of MoE architectures, present common pitfalls such as expert collapse and load imbalance, and summarize the most prominent strategies proposed to stabilize and improve training.

5.1 Challenges in Training MoE Models

Expert Collapse. One of the most persistent issues in MoE training is expert collapse, where the gating network disproportionately favors a small subset of experts [85]. This leads to under-utilization of the model's full capacity, reduces specialization, and undermines the benefits of modularity. Collapse is especially prevalent in sparse MoE setups where only a few experts are activated per input. Once a dominant routing pattern emerges, experts that are rarely selected may receive few or no gradient updates, leading to a self-reinforcing feedback loop.

Load Imbalance. Closely related to expert collapse is the problem of load imbalance [86]. Ideally, the routing mechanism should distribute the computational load evenly across experts. However, without explicit regularization, the gating function may learn to favor a small number of experts for most inputs, creating bottlenecks during training and inference. In distributed training settings, this imbalance also leads to poor utilization of compute resources and degraded throughput.

Sparse Gradients. Sparse activation implies that only a subset of experts are updated during each training step [87]. While this improves computational efficiency, it also means that expert networks receive fewer updates per epoch compared to the rest of the model [88]. This can slow convergence and reduce learning effectiveness, particularly for experts that are seldom activated [89].

Non-differentiable Routing. Many routing mechanisms in sparse MoE architectures rely on discrete top-k selection, which introduces non-differentiability into the model [90]. This complicates backpropagation and makes gradient-based optimization less straightforward [91]. Approximation techniques such as soft top-k or noisy gating are often required to enable end-to-end learning [92].

5.2 Stabilization Techniques

Auxiliary Load Balancing Loss. One widely adopted solution to expert collapse and imbalance is the inclusion of an auxiliary loss that encourages uniform

expert usage [93]. For instance, the GShard [33] loss is defined as:

$$\mathcal{L}_{\text{balance}} = C \cdot \text{Coeff} \cdot \left(\frac{(\sum_i p_i)^2}{\sum_i p_i^2 + \epsilon} \right), \tag{4}$$

where p_i denotes the cumulative gating probability for expert i, C is a normalizing constant, and ϵ prevents division by zero. This objective is combined with the main task loss to promote load diversity without compromising performance [94].

Noise Injection. Shazeer et al. [22] proposed injecting Gaussian noise into the gating logits during training. This stochastic perturbation encourages exploration in the routing space and prevents premature convergence to degenerate routing patterns [95]. The gating scores become:

$$g_i(\mathbf{x}) = \frac{\exp((W_g\mathbf{x} + \mathcal{N}(0, \sigma^2))_i)}{\sum_j \exp((W_g\mathbf{x} + \mathcal{N}(0, \sigma^2))_j)}, \tag{5}$$

where $\mathcal{N}(0, \sigma^2)$ is Gaussian noise with standard deviation σ.

Capacity Constraints. To prevent expert overload, many MoE models impose a capacity limit on each expert during routing. Inputs that would exceed an expert's capacity are either rerouted to lower-ranked experts or dropped [96]. This mechanism ensures that no single expert dominates computation and that all experts contribute to the training process [97].

Top-k Routing with Prioritized Assignment. Some models use top-k routing but add prioritization heuristics to handle overflow [98]. In GShard, for example, tokens are assigned to experts based on gating scores, but only the highest-scoring tokens are retained when capacity is exceeded. This preserves high-confidence assignments while respecting capacity limits.

5.3 Training Strategies

Distributed Training and Expert Parallelism. Given the large number of parameters and the conditional nature of computation, training MoE models efficiently often requires distributed computation [15]. Expert parallelism [33] is a strategy where different experts are distributed across different devices or nodes, enabling scalable training. However, this setup necessitates careful management of communication overhead and synchronization.

Token Dropping and Gradient Skipping. To further enhance training throughput, some implementations allow low-gated tokens to be dropped (not processed) and prevent their gradients from backpropagating through the gating network. This reduces memory footprint and computational cost, particularly during the early stages of training when gating distributions are still noisy [99].

Warm-up Schedules and Expert Freezing. A gradual warm-up schedule for gating sparsity and auxiliary losses can improve convergence. Additionally, some models temporarily freeze gating networks or expert parameters to stabilize learning before fine-tuning the entire system end-to-end.

Gradient Normalization and Balancing. Sparse MoE architectures suffer from uneven gradient magnitudes due to irregular expert utilization. Normalization techniques that rescale gradients based on expert usage statistics can mitigate these issues and improve training stability.

5.4 Recent Advances in MoE Optimization

Recent research has proposed advanced training methodologies that further enhance the robustness of MoE systems. For example, reinforcement learning-based gating [**?**] introduces a reward signal to improve expert routing decisions, while differentiable approximations to hard selection (e.g., Gumbel-Softmax) allow for smoother optimization [100]. There are also efforts to integrate meta-learning and continual learning frameworks to dynamically adapt expert specializations over time, thus making MoE models more flexible and lifelong-learning capable [101].

5.5 Summary

Training Mixture of Experts models involves navigating a complex interplay between model sparsity, routing decisions, expert diversity, and optimization stability. The innovations developed to address these challenges have not only made large-scale MoE training feasible, but have also provided valuable insights for other sparse and modular learning paradigms [102]. In the next section, we examine the broad range of applications where MoE models have demonstrated competitive or state-of-the-art performance, highlighting their versatility and practical impact [103].

6 Applications of Mixture of Experts

Mixture of Experts (MoE) models have gained widespread traction across a diverse set of application domains, demonstrating state-of-the-art performance in many large-scale tasks. The core strengths of MoE—scalability, modularity, and conditional computation—make them especially well-suited for environments characterized by high data heterogeneity, task complexity, and resource constraints [104]. In this section, we survey the prominent application areas where MoE architectures have been successfully deployed, discuss the empirical benefits observed, and highlight open questions around generalization and transferability [22].

6.1 Natural Language Processing

Natural Language Processing (NLP) has been the most prominent application area for modern MoE models. The breakthrough came with the integration of sparse MoE layers into transformer-based architectures, which allowed for unprecedented scaling of language models.

Machine Translation. GShard [33] was among the first large-scale MoE-based models applied to machine translation. It introduced sparsely-gated MoE layers into Transformer encoders and decoders, leading to significant improvements in BLEU scores while maintaining manageable computational costs. The capacity to scale to billions of parameters without a proportional increase in computation made GShard a milestone in efficient large-model training [105].

Language Modeling. The Switch Transformer [26], a sparsely-gated model with top-1 expert routing, achieved state-of-the-art results on multiple language modeling benchmarks such as C4, WikiText-103, and LAMBADA, while being highly efficient during training and inference [106]. Similarly, GLaM [27] pushed the envelope further by scaling up to hundreds of billions of parameters and introducing enhanced routing and expert regularization. These models showed that expert sparsity can enable models to train faster and generalize better with fewer FLOPs per parameter [107].

Multilingual and Multitask Models. MoE has also enabled the creation of unified multilingual models that can scale effectively across languages [108]. For instance, the M6-T model [**?**] uses MoE in a multitask setting to jointly learn from different modalities and languages, demonstrating strong cross-lingual transfer. In multitask NLP, MoE allows selective routing of different tasks to different experts, leading to improved task specialization and better resource sharing.

6.2 Computer Vision

Although MoE originated in NLP applications, its modular and scalable nature has also proven valuable in the vision domain, particularly in the context of vision transformers and large-scale image classification [109].

Vision Transformers with MoE. The Vision MoE (V-MoE) model [31] adapts sparse MoE layers into the feed-forward blocks of vision transformers [110]. V-MoE achieves improved performance on ImageNet and other vision benchmarks by introducing conditional computation, enabling the model to dynamically adjust its capacity based on input complexity [111]. It also shows that routing decisions made on intermediate representations can correlate with semantic content, enabling some degree of interpretability [112].

Efficient Training of Large-Scale Vision Models. MoE has been used to reduce training costs in large-scale vision systems by allowing only a subset of experts to be active at each step [113]. Conditional computation improves efficiency and allows deployment of much larger models than would be feasible under full computation regimes [114].

6.3 Speech and Audio Processing

MoE models have also been successfully employed in speech recognition and audio classification, where data exhibits temporal variability and task diversity [115].

Self-Supervised Speech Models. Wu et al. [**?**] incorporated MoE into their self-supervised speech representation learning framework (BigSSL). They showed that sparse experts can lead to higher representational capacity and downstream task performance in phoneme recognition and speaker identification.

Multi-Speaker and Multi-Task Scenarios. MoE is particularly useful in multi-speaker environments, where different experts can specialize in different speaker profiles or dialects. This modularity improves generalization across demographic groups and enhances fairness in speech applications [116].

6.4 Multi-Modal Learning

The conditional and modular nature of MoE makes it a natural fit for multi-modal models that integrate inputs from text, vision, audio, and other modalities [117].

Cross-Modal Routing. Goyal et al. [**?**] proposed a multi-modal MoE architecture in which different experts are selectively activated based on the modality of the input [118]. This allows for shared experts across modalities while also supporting modality-specific specializations. Such architectures are beneficial in tasks like image captioning, visual question answering (VQA), and cross-modal retrieval.

Scalable Multi-Modal Transformers. MoE has been used to scale up multi-modal transformers for tasks such as document understanding, where models must process textual, visual, and structural information simultaneously. Conditional activation allows these models to dynamically allocate resources depending on the complexity of each input component.

6.5 Continual, Lifelong, and Meta-Learning

The inherent modularity of MoE models also makes them appealing in continual and lifelong learning scenarios, where models must adapt to new tasks without catastrophic forgetting.

Expert Growth and Replacement. Recent works have explored dynamic expert creation, pruning, and replacement strategies within MoE systems. As new tasks arrive, the model can instantiate new experts or repurpose underutilized ones, facilitating continual adaptation [27].

Task-Aware Routing. In meta-learning contexts, MoE routing functions can learn to specialize experts based on task embeddings, effectively partitioning model capacity across task distributions [119]. This enables few-shot and zero-shot generalization without retraining the entire model.

6.6 Industry Applications and Deployment

MoE models have been deployed in large-scale industrial systems due to their favorable trade-offs between performance and cost [120]. Examples include:

- Google's production translation systems based on GShard and Switch Transformer.
- YouTube's recommendation engines incorporating MoE-style modularity for personalization.
- Vision systems for content moderation and retrieval using V-MoE for scalable classification [121].

These real-world deployments highlight MoE's practical viability and its ability to balance model scale, inference latency, and accuracy under production constraints [122].

6.7 Summary

Mixture of Experts models have shown strong empirical success across a wide range of application domains, particularly where scalability, efficiency, and adaptability are essential. Their versatility makes them appealing not only for research but also for real-world deployment in industry. Despite their promising results, there remain open challenges in generalization, task transfer, and interpretability [123]. In the next section, we review the theoretical underpinnings of MoE models and explore recent work that attempts to formalize their behavior and advantages from a principled perspective [124].

7 Conclusion

Mixture of Experts (MoE) has re-emerged as a powerful and scalable paradigm in modern deep learning, offering a compelling framework for conditional computation, modular specialization, and efficient scaling. Over the past few years, the evolution of MoE models has transitioned from theoretical interest to practical dominance, with applications spanning natural language processing, computer vision, speech processing, and multi-modal learning. By decoupling capacity from computational cost, MoE enables the training of ultra-large models while maintaining tractability in both training and inference.

This survey has comprehensively examined the foundations, methodologies, and frontiers of MoE architectures. We began with an overview of their conceptual origins and fundamental design, distinguishing between sparse and dense expert configurations, hard and soft routing schemes, and hierarchical extensions. We then explored the challenges of training MoE systems—highlighting issues like expert collapse, load imbalance, and routing non-differentiability—along with the techniques proposed to stabilize and optimize them. A deep dive into applications demonstrated MoE's versatility across domains, from high-performance language models like Switch Transformer and GLaM, to scalable vision systems like V-MoE, to dynamic, modular architectures in multi-modal and continual learning settings.

On the theoretical front, we reviewed how MoE models relate to and extend principles from ensemble learning, conditional computation, and modularity theory. Recent work has begun to formalize the generalization and expressivity of MoE, but much remains to be understood. The dynamic nature of expert routing, the complex optimization landscape, and the combinatorial nature of sparse

activation present rich avenues for continued research.

Despite their promise, MoE models are not without limitations. Challenges remain in terms of expert under-utilization, scaling efficiency in distributed environments, robustness to routing errors, and interpretability of expert specialization. Furthermore, the lack of standardized benchmarks and reproducibility practices poses a barrier to fair comparison and wider adoption. Future work must address these concerns while also exploring new frontiers such as dynamic expert generation, lifelong learning, energy-efficient deployment, and integration with emerging paradigms like retrieval-augmented models and foundation models.

In summary, Mixture of Experts represents a paradigm shift in model architecture design—moving from monolithic networks to modular, adaptive systems. Its success demonstrates that computation need not be uniform to be powerful. As research continues to mature, MoE is poised to play a central role in the next generation of intelligent systems, combining scalability, flexibility, and specialization in a single unifying framework.

References

[1] William Fedus, Jeff Dean, and Barret Zoph. A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667*, 2022.

[2] Shwai He, Run-Ze Fan, Liang Ding, Li Shen, Tianyi Zhou, and Dacheng Tao. Merging experts into one: Improving computational efficiency of mixture of experts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14685–14691, 2023.

[3] Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023.

[4] Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. Openmoe: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2402.01739*, 2024.

[5] Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2021.

[6] Yassine Zniyed, Thanh Phuong Nguyen, et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[7] Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. MixLoRA: Enhancing Large Language Models Fine-Tuning with LoRA based Mixture of Experts. *arXiv preprint arXiv:2404.15159*, 2024.

[8] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13041–13049, 2020.

[9] Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, et al. Mixture-of-experts meets instruction tuning: A winning combination for large language models. *arXiv preprint arXiv:2305.14705*, 2023.

[10] Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022.

[11] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[13] Bowen Pan, Yikang Shen, Haokun Liu, Mayank Mishra, Gaoyuan Zhang, Aude Oliva, Colin Raffel, and Rameswar Panda. Dense Training, Sparse Inference: Rethinking Training of Mixture-of-Experts Language Models. *arXiv preprint arXiv:2404.05567*, 2024.

[14] Dheeru Dua, Shruti Bhosale, Vedanuj Goswami, James Cross, Mike Lewis, and Angela Fan. Tricks for Training Sparse Translation Models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3340–3345, 2022.

[15] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[16] Tianlong Chen, Zhenyu Zhang, AJAY KUMAR JAISWAL, Shiwei Liu, and Zhangyang Wang. Sparse MoE as the New Dropout: Scaling Dense and Self-Slimmable Transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

[17] Liang Shen, Zhihua Wu, WeiBao Gong, Hongxiang Hao, Yangfan Bai, HuaChao Wu, Xinxuan Wu, Jiang Bian, Haoyi Xiong, Dianhai Yu, et al. Se-moe: A scalable and efficient mixture-of-experts distributed training and inference system. *arXiv preprint arXiv:2205.10034*, 2022.

[18] Xin Wang, Fisher Yu, Lisa Dunlap, Yi-An Ma, Ruth Wang, Azalia Mirhoseini, Trevor Darrell, and Joseph E Gonzalez. Deep mixture of experts via shallow embedding. In *Uncertainty in artificial intelligence*, pages 552–562. PMLR, 2020.

[19] Yunhao Gou, Zhili Liu, Kai Chen, Lanqing Hong, Hang Xu, Aoxue Li, Dit-Yan Yeung, James T Kwok, and Yu Zhang. Mixture of cluster-conditional lora experts for vision-language instruction tuning. *arXiv preprint arXiv:2312.12379*, 2023.

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[21] Wuyang Chen, Yanqi Zhou, Nan Du, Yanping Huang, James Laudon, Zhifeng Chen, and Claire Cui. Lifelong language pretraining with distribution-specialized experts. In *International Conference on Machine Learning*, pages 5383–5395. PMLR, 2023.

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

[23] Qwen Team. Qwen1.5-MoE: Matching 7B Model Performance with 1/3 Activated Parameters", February 2024. URL `https://qwenlm.github.io/blog/qwen-moe/`.

[24] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024.

[25] Xiaonan Nie, Pinxue Zhao, Xupeng Miao, Tong Zhao, and Bin Cui. HetuMoE: An efficient trillion-scale mixture-of-expert distributed training system. *arXiv preprint arXiv:2203.14685*, 2022.

[26] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[27] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022.

[28] Shawn Tan, Yikang Shen, Rameswar Panda, and Aaron Courville. Scattered Mixture-of-Experts Implementation. *arXiv preprint arXiv:2403.08245*, 2024.

[29] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[30] Snowflake AI Research Team. Snowflake Arctic: The Best LLM for Enterprise AI — Efficiently Intelligent, Truly Open, April 2024. URL `https://www.snowflake.com/blog/arctic-open-efficient-foundation-language-models-snowflake/`.

[31] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021.

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[33] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen.

Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[34] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[35] Jiaao He, Jidong Zhai, Tiago Antunes, Haojie Wang, Fuwen Luo, Shangfeng Shi, and Qin Li. Fastermoe: modeling and optimizing training of large-scale dynamic pre-trained models. In *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 120–134, 2022.

[36] Kang Min Yoo, Jaegeun Han, Sookyo In, Heewon Jeon, Jisu Jeong, Jaewook Kang, Hyunwook Kim, Kyung-Min Kim, Munhyong Kim, Sungju Kim, et al. HyperCLOVA X Technical Report. *arXiv preprint arXiv:2404.01954*, 2024.

[37] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

[38] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[39] Siddharth Singh, Olatunji Ruwase, Ammar Ahmad Awan, Samyam Rajbhandari, Yuxiong He, and Abhinav Bhatele. A Hybrid Tensor-Expert-Data Parallelism Approach to Optimize Mixture-of-Experts Training. In *Proceedings of the 37th International Conference on Supercomputing*, pages 203–214, 2023.

[40] Andrew Davis and Itamar Arel. Low-rank approximations for conditional feedforward computation in deep neural networks. *arXiv preprint arXiv:1312.4461*, 2013.

[41] Xiaofeng Zhang, Yikang Shen, Zeyu Huang, Jie Zhou, Wenge Rong, and Zhang Xiong. Mixture of Attention Heads: Selecting Attention Heads Per Token. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4150–4162, 2022.

[42] Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Large memory layers with product keys. *Advances in Neural Information Processing Systems*, 32, 2019.

[43] Do Huu Dat, Po Yuan Mao, Tien Hoang Nguyen, Wray Buntine, and Mohammed Bennamoun. 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.

[44] Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[45] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[46] Mingshu Zhai, Jiaao He, Zixuan Ma, Zan Zong, Runqing Zhang, and Jidong Zhai. {SmartMoE}: Efficiently Training {Sparsely-Activated} Models through Combining Offline and Online Parallelization. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pages 961–975, 2023.

[47] Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. Adamoe: Token-adaptive routing with null experts for mixture-of-experts language models. *arXiv preprint arXiv:2406.13233*, 2024.

[48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[49] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015.

[50] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32, 2019.

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

[52] Shwai He, Liang Ding, Daize Dong, Boan Liu, Fuqiang Yu, and Dacheng Tao. Pad-net: An efficient framework for dynamic networks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14354–14366, 2023.

[53] An Yang, Junyang Lin, Rui Men, Chang Zhou, Le Jiang, Xianyan Jia, Ang Wang, Jie Zhang, Jiamang Wang, Yong Li, et al. M6-t: Exploring sparse expert models and beyond. *arXiv preprint arXiv:2105.15082*, 2021.

[54] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[55] Amjad Almahairi, Nicolas Ballas, Tim Cooijmans, Yin Zheng, Hugo Larochelle, and Aaron Courville. Dynamic capacity networks. In *International Conference on Machine Learning*, pages 2549–2558. PMLR, 2016.

[56] Qinyuan Ye, Juan Zha, and Xiang Ren. Eliciting and Understanding Cross-task Skills with Task-level Mixture-of-Experts. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2567–2592, 2022.

[57] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[58] Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Scott Yih, and Madian Khabsa. UniPELT: A Unified Framework for Parameter-Efficient Language Model Tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6253–6264, 2022.

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

[60] Juyong Jiang, Peiyan Zhang, Yingtao Luo, Chaozhuo Li, Jae Boum Kim, Kai Zhang, Senzhang Wang, Xing Xie, and Sunghun Kim. AdaMCT: adaptive mixture of CNN-transformer for sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 976–986, 2023.

[61] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

[62] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*, 2021.

[63] OpenAI. Chatgpt: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt`, 2022.

[64] Ruisi Cai, Saurav Muralidharan, Greg Heinrich, Hongxu Yin, Zhangyang Wang, Jan Kautz, and Pavlo Molchanov. Flextron: Many-in-one flexible large language model. In *Forty-first International Conference on Machine Learning*.

[65] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*, 2022.

[66] Yun Zhu, Nevan Wichers, Chu-Cheng Lin, Xinyi Wang, Tianlong Chen, Lei Shu, Han Lu, Canoee Liu, Liangchen Luo, Jindong Chen, et al. Sira: Sparse mixture of low rank adaptation. *arXiv preprint arXiv:2311.09179*, 2023.

[67] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2021.

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

[69] Shawn Tan, Yikang Shen, Zhenfang Chen, Aaron Courville, and Chuang Gan. Sparse Universal Transformer. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 169–179, 2023.

[70] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning. In *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)*, pages 7319–7328, 2021.

[71] Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, et al. Tutel: Adaptive

mixture-of-experts at scale. *Proceedings of Machine Learning and Systems*, 5, 2023.

[72] Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, and Yuxiong He. Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. In *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pages 1–14, 2021.

[73] Qwen Team. Introducing Qwen1.5, February 2024. URL `https://qwenlm.github.io/blog/qwen1.5/`.

[74] Yassine Zniyed, Thanh Phuong Nguyen, et al. Efficient tensor decomposition-based filter pruning. *Neural Networks*, 178:106393, 2024.

[75] Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. MoEfication: Transformer Feed-forward Layers are Mixtures of Experts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 877–890, 2022.

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

[77] Stephen Roller, Sainbayar Sukhbaatar, Jason Weston, et al. Hash layers for large sparse models. *Advances in Neural Information Processing Systems*, 34:17555–17566, 2021.

[78] Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Jianfeng Gao, and Tuo Zhao. Taming Sparsely Activated Transformer with Stochastic Experts. In *International Conference on Learning Representations*, 2021.

[79] Xiaonan Nie, Xupeng Miao, Zilong Wang, Zichao Yang, Jilong Xue, Lingxiao Ma, Gang Cao, and Bin Cui. Flexmoe: Scaling large-scale sparse pre-trained model training via dynamic device placement. *Proceedings of the ACM on Management of Data*, 1(1):1–19, 2023.

[80] Yunshui Li, Binyuan Hui, ZhiChao Yin, Min Yang, Fei Huang, and Yongbin Li. Pace: Unified multi-modal dialogue pre-training with progressive and compositional experts. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13402–13416, 2023.

[81] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 269–278, 2020.

[82] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

[83] Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A Smith, and Luke Zettlemoyer. DEMix Layers: Disentangling Domains for Modular Language Modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, 2022.

[84] Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. {Zero-offload}: Democratizing {billion-scale} model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 551–564, 2021.

[85] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

[86] Mohammed Muqeeth, Haokun Liu, and Colin Raffel. Soft merging of experts with adaptive routing. *arXiv preprint arXiv:2306.03745*, 2023.

[87] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[88] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

[89] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems*, 35:9564–9576, 2022.

[90] Chang Chen, Min Li, Zhihua Wu, Dianhai Yu, and Chao Yang. Ta-moe: Topology-aware large scale mixture-of-expert training. *Advances in Neural Information Processing Systems*, 35:22173–22186, 2022.

[91] Fuzhao Xue, Xiaoxin He, Xiaozhe Ren, Yuxuan Lou, and Yang You. One student knows all experts know: From sparse to dense. *arXiv preprint arXiv:2201.10890*, 2022.

[92] Ningxin Zheng, Huiqiang Jiang, Quanlu Zhang, Zhenhua Han, Lingxiao Ma, Yuqing Yang, Fan Yang, Chengruidong Zhang, Lili Qiu, Mao Yang, et al. Pit: Optimization of dynamic sparse deep learning models via permutation invariant transformation. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 331–347, 2023.

[93] Zixuan Ma, Jiaao He, Jiezhong Qiu, Huanqi Cao, Yuanwei Wang, Zhenbo Sun, Liyan Zheng, Haojie Wang, Shizhi Tang, Tianyu Zheng, et al. BaGuaLu: targeting brain scale pretrained models with over 37 million cores. In *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 192–204, 2022.

[94] Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. *arXiv preprint arXiv:2309.05444*, 2023.

[95] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[96] Tianyu Chen, Shaohan Huang, Yuan Xie, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. Task-specific expert pruning for sparse mixture-of-experts. *arXiv preprint arXiv:2206.00277*, 2022.

[97] Databricks. Introducing DBRX: A New State-of-the-Art Open LLM, March 2024. URL `https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm`.

[98] Fuzhao Xue, Ziji Shi, Futao Wei, Yuxuan Lou, Yong Liu, and Yang You. Go wider instead of deeper. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8779–8787, 2022.

[99] Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. StableMoE: Stable Routing Strategy for Mixture of Experts.

In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7085–7095, 2022.

[100] Yongqi Huang, Peng Ye, Xiaoshui Huang, Sheng Li, Tao Chen, and Wanli Ouyang. Experts weights averaging: A new general training scheme for vision transformers. *arXiv preprint arXiv:2308.06093*, 2023.

[101] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[102] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3366–3375, 2017.

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

[104] Szymon Antoniak, Sebastian Jaszczur, Michał Krutul, Maciej Pióro, Jakub Krajewski, Jan Ludziejewski, Tomasz Odrzygóźdź, and Marek Cygan. Mixture of Tokens: Efficient LLMs through Cross-Example Aggregation. *arXiv preprint arXiv:2310.15961*, 2023.

[105] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[106] Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.

[107] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. In *International conference on machine learning*, pages 18332–18346. PMLR, 2022.

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

[109] Shaoxiang Chen, Zequn Jie, and Lin Ma. Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms. *arXiv preprint arXiv:2401.16160*, 2024.

[110] Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R Devanur, Gregory R Ganger, Phillip B Gibbons, and Matei Zaharia. PipeDream: generalized pipeline parallelism for DNN training. In *Proceedings of the 27th ACM symposium on operating systems principles*, pages 1–15, 2019.

[111] Hongyi Wang, Felipe Maia Polo, Yuekai Sun, Souvik Kundu, Eric Xing, and Mikhail Yurochkin. Fusing Models with Complementary Expertise. In *The Twelfth International Conference on Learning Representations*, 2023.

[112] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022.

[113] Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. Revisiting Parameter-Efficient Tuning: Are We Really There Yet? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2612–2626, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.168. URL `https://aclanthology.org/2022.emnlp-main.168`.

[114] Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. Higher Layers Need More LoRA Experts. *arXiv preprint arXiv:2402.08562*, 2024.

[115] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

[116] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[117] Young Jin Kim, Ammar Ahmad Awan, Alexandre Muzio, Andres Felipe Cruz Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, and Hany Hassan Awadalla. Scalable and efficient moe training for multitask multilingual models. *arXiv preprint arXiv:2109.10465*, 2021.

[118] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[119] xAI. Grok-1, March 2024. URL `https://github.com/xai-org/grok-1`.

[120] Yunxin Li, Shenyuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. Uni-MoE: Scaling Unified Multimodal LLMs with Mixture of Experts. *arXiv preprint arXiv:2405.11273*, 2024.

[121] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[122] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5579–5588, 2021.

[123] Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. Branch-Train-Merge: Embarrassingly Parallel Training of Expert Language Models. In *First Workshop on Interpolation Regularizers and Beyond at NeurIPS 2022*, 2022.

[124] Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, et al. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*, 2023.