Article

Not peer-reviewed version

Improving Deep Learning Performance with Mixture of Experts and Sparse Activation

Jaye Nnamdi , Vasily Dimitri * , Somerled Amar

Posted Date: 10 March 2025

doi: 10.20944/preprints202503.0611.v1

Keywords: Mixture of Experts; Deep Learning; Neural Networks; Model Scalability; Sparse Activation; Modular Architectures; Expert Routing; Natural Language Processing; Computer Vision; Recommendation Systems; Training Efficiency; Interpretability; Fairness in AI

 doi:10.20944/preprints202503.0611.v1

Article

Improving Deep Learning Performance with Mixture of Experts and Sparse Activation

Jaye Nnamdi, Vasily Dimitri * and Somerled Amar

King Abdullah University of Science and Technology
* Correspondence: vasily.dimitri@kaust.edu.sa

Abstract: The increasing complexity and scale of modern machine learning models have led to growing computational demands, raising concerns about efficiency, scalability, and adaptability. Traditional deep learning architectures often struggle to balance computational cost with model expressiveness, particularly in tasks requiring specialization across diverse data distributions. One promising solution is the use of modular architectures that allow selective activation of parameters, enabling efficient resource allocation while maintaining high performance. Mixture of Experts (MoE) is a widely adopted modular approach that partitions the model into multiple specialized experts, dynamically selecting a subset of them for each input. This technique has demonstrated remarkable success in large-scale machine learning applications, including natural language processing, computer vision, speech recognition, and recommendation systems. By leveraging sparse activation, MoE architectures achieve significant computational savings while scaling to billions of parameters. This survey provides a comprehensive overview of MoE, covering its fundamental principles, architectural variations, training strategies, and key applications. Additionally, we discuss the major challenges associated with MoE, including training stability, expert imbalance, interpretability, and hardware constraints. Finally, we explore potential future research directions aimed at improving efficiency, fairness, and real-world deployability. As machine learning continues to advance, MoE is poised to play a crucial role in the development of scalable and adaptive AI systems.

Keywords: mixture of experts; deep learning; neural networks; model scalability; sparse activation; modular architectures; expert routing; natural language processing; computer vision; recommendation systems; training efficiency; interpretability; fairness in AI

1. Introduction

In recent years, machine learning has witnessed remarkable advancements, leading to the development of increasingly sophisticated models capable of handling complex tasks across various domains [1]. From natural language processing and computer vision to robotics and healthcare, these models have demonstrated unprecedented capabilities in understanding and processing vast amounts of data [2]. However, as machine learning models continue to grow in size and complexity, concerns related to computational efficiency, scalability, and generalization have become more prominent. This has motivated researchers to explore alternative architectures that can efficiently allocate resources while maintaining high predictive performance. One promising approach to addressing these challenges is the Mixture of Experts (MoE) framework, a concept rooted in the divide-and-conquer principle [3]. The core idea behind MoE is to decompose a complex problem into smaller, more manageable subproblems and assign specialized models, known as "experts," to handle different regions of the input space. A gating mechanism dynamically selects the most relevant experts for a given input, enabling the model to distribute computational resources effectively. By activating only a subset of experts for each input, MoE can significantly reduce computational costs compared to traditional monolithic models, making it particularly appealing for large-scale machine learning applications [4]. The MoE paradigm has a long history in machine learning, dating back to early work in neural

 doi:10.20944/preprints202503.0611.v1

networks and ensemble learning. Initially proposed as a method for improving model interpretability and robustness, MoE has evolved substantially, especially with the advent of deep learning. Recent advancements in deep MoE architectures have led to state-of-the-art performance in tasks such as language modeling, image recognition, and reinforcement learning [5]. Large-scale implementations of MoE, such as those used in natural language processing, have demonstrated that models with trillions of parameters can achieve superior performance while maintaining efficiency by leveraging sparse computation [6]. Despite its advantages, MoE presents several challenges that must be addressed to unlock its full potential [7]. The training of MoE models introduces additional complexity, as the gating mechanism must learn to effectively allocate inputs to experts while preventing issues such as expert over-specialization and mode collapse [8]. Furthermore, ensuring load balancing among experts is crucial to avoid scenarios where certain experts dominate the learning process while others remain underutilized. In addition, MoE architectures pose unique challenges in terms of memory efficiency, hardware acceleration, and deployment in real-world applications [9]. This survey provides a comprehensive review of the Mixture of Experts framework, covering its theoretical foundations, various architectural designs, and training methodologies. We explore different strategies for expert selection and gating functions, analyze their advantages and limitations, and discuss recent breakthroughs in deep MoE architectures. Additionally, we examine the practical applications of MoE across different domains, highlighting its impact on large-scale machine learning. Finally, we discuss open challenges and future research directions, providing insights into how MoE can continue to evolve as a key paradigm in modern artificial intelligence.

2. Background and Theoretical Foundations

The Mixture of Experts (MoE) framework is grounded in the principle of divide-and-conquer, where complex learning tasks are decomposed into simpler subproblems, each handled by specialized models [10]. This concept has its roots in ensemble learning, modular neural networks, and probabilistic modeling, drawing from theories in optimization and statistical learning.

2.1. Historical Perspective

The idea of MoE was first introduced in the early 1990s by Jacobs et al. [11], who proposed a modular neural network architecture where different subnetworks (experts) were trained to specialize in different parts of the input space. A gating network, trained alongside the experts, was responsible for dynamically weighting their outputs based on the input data. This approach was motivated by the observation that a single model might struggle to generalize across diverse data distributions, whereas a collection of specialized models could improve both efficiency and interpretability. Over the years, MoE has evolved through various adaptations, benefiting from advances in deep learning and large-scale optimization [12]. While early implementations of MoE were primarily used for small-scale tasks, the recent resurgence of interest in sparse neural architectures has led to the development of highly scalable MoE models deployed in state-of-the-art systems such as large language models.

2.2. Mathematical Formulation

Formally, a Mixture of Experts model consists of a set of K expert networks $\{E_1, E_2, \ldots, E_K\}$ and a gating function $G(x)$ that assigns a weight to each expert based on the input x [13]. The model output is computed as a weighted sum of the expert outputs:

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

where $G_i(x)$ represents the gating function's output for expert E_i. The gating function is typically implemented as a softmax layer:

 doi:10.20944/preprints202503.0611.v1

$$G_i(x) = \frac{\exp(W_i^T x)}{\sum_{j=1}^{K} \exp(W_j^T x)}, \tag{2}$$

where W_i are the learnable parameters of the gating network. This formulation ensures that the gating values sum to one, effectively acting as a probability distribution over the experts [14].

2.3. Training Strategies

Training an MoE model involves optimizing both the expert networks and the gating function. The objective function typically consists of a loss term that encourages correct predictions and a regularization term to prevent mode collapse and promote balanced expert utilization [15]. Standard training strategies include:

- **Hard and Soft Gating:** Hard gating assigns each input to a single expert, while soft gating allows multiple experts to contribute to the final output [16]. Soft gating is more flexible but computationally expensive [17].
- **Load Balancing:** To ensure all experts contribute meaningfully, additional loss terms are often introduced to encourage balanced expert usage [18].
- **Gradient Routing:** Efficient gradient flow through the gating mechanism is crucial to ensure stable training and avoid expert underutilization.

2.4. Comparison with Other Architectures

MoE differs from traditional ensemble learning methods, such as bagging and boosting, in that experts are dynamically selected during inference rather than being aggregated in a static manner. Compared to conventional deep networks, MoE enables sparse computation, activating only a subset of parameters per input. This provides significant efficiency gains, particularly in large-scale models [19–21].

2.5. Challenges and Limitations

Despite its advantages, MoE presents several challenges, including:

- **Expert Specialization and Mode Collapse:** Some experts may become over-specialized or completely inactive, reducing the model's diversity.
- **Computational Complexity:** Although sparse activation reduces computation, routing decisions introduce overhead.
- **Scalability Issues:** Training large-scale MoE models requires careful optimization strategies to prevent communication bottlenecks in distributed environments [22].

The next sections will explore different architectural designs and training methodologies that have been proposed to address these challenges and enhance the effectiveness of MoE models [23].

3. Architectural Variants of Mixture of Experts

The Mixture of Experts (MoE) framework has evolved significantly since its inception, leading to various architectural designs tailored to different applications [24]. These architectures differ in how experts are organized, how the gating mechanism operates, and how computational resources are allocated. This section explores several key MoE architectures, highlighting their strengths, weaknesses, and use cases [25].

3.1. Classic MoE Architecture

The traditional MoE architecture consists of a set of K expert networks and a gating network that assigns input-dependent weights to each expert. The model output is computed as a weighted sum of the expert outputs:

 doi:10.20944/preprints202503.0611.v1

$$y = \sum_{i=1}^{K} G_i(x) E_i(x), \tag{3}$$

where $G_i(x)$ is the gating function's output for expert E_i. The gating mechanism is typically implemented as a softmax function, ensuring that the weights sum to one [26]. While this architecture is effective in capturing diverse data patterns, it has certain drawbacks, including potential expert underutilization and scalability challenges [27]. Training stability is another concern, as an unbalanced gating mechanism may lead to some experts dominating while others remain inactive.

3.2. Hierarchical MoE

Hierarchical MoE (HMoE) introduces multiple layers of experts, where each gating network not only selects experts at its level but also routes decisions to deeper layers [28]. This structure allows for increased specialization while maintaining computational efficiency [29]. **Advantages:**

- Improved model capacity and flexibility by enabling hierarchical decision-making [30].
- Better scalability as experts at different levels specialize in finer-grained subproblems.
- Potential for more structured representations, making the model more interpretable.

Challenges:

- Increased training complexity due to multiple gating functions.
- Potential vanishing gradient issues in deep hierarchies.

3.3. Sparse MoE with Top-K Gating

A major limitation of traditional MoE is its computational cost when all experts contribute to each inference step. Sparse MoE mitigates this by activating only a small subset (e.g., the top-k most relevant) of experts for each input [31]. This is often implemented using a *Top-K gating function*, where only the top-k experts with the highest gating scores receive nonzero weights:

$$G_i(x) = \begin{cases} \frac{\exp(W_i^T x)}{\sum_{j \in \mathcal{S}} \exp(W_j^T x)} & \text{if } i \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

where $\mathcal{S}$ represents the set of top-k experts chosen for the given input. **Advantages:**

- Significant reduction in computational cost as only a subset of experts are used per input.
- Improved efficiency for large-scale models, particularly in distributed environments.
- Reduced risk of expert overfitting, as sparsity encourages better generalization.

Challenges:

- Load balancing issues, where certain experts may receive disproportionately more inputs than others [32].
- Increased variance in training, as fewer experts contribute to each update.

3.4. Soft MoE vs [33]. Hard MoE

MoE architectures can be broadly classified into soft and hard gating mechanisms:
Soft MoE:

- Uses a smooth softmax gating function to assign continuous weights to all experts.
- Allows for multiple experts to contribute to each prediction [34].
- Generally results in more stable training but requires more computation.

Hard MoE:

- Selects a discrete subset (often one) of experts per input, making inference more efficient [35].
- Can be implemented using techniques such as hard thresholding or reinforcement learning-based routing.

 doi:10.20944/preprints202503.0611.v1

- More challenging to train due to non-differentiability, requiring techniques such as the Gumbel-Softmax trick.

3.5. Recent Advances in MoE Architectures

Recent research has introduced several improvements to MoE architectures, addressing key limitations such as load balancing, computational efficiency, and robustness:

- **Switch Transformers:** Introduced by Fedus et al. [30], this model simplifies MoE by using only one active expert per input token, significantly reducing computation while maintaining high performance in language modeling tasks [36].
- **Routing Networks:** Dynamic routing strategies, such as reinforcement learning-based gating, have been explored to optimize expert selection based on long-term learning objectives.
- **Distillation-Augmented MoE:** Some architectures leverage knowledge distillation to enhance expert generalization and reduce redundancy among experts.

3.6. Comparison of MoE Architectures

Table 1 provides a comparative summary of the discussed MoE architectures in terms of computational efficiency, specialization, and scalability.

Table 1. Comparison of Different MoE Architectures.

Architecture	Computational Cost	Specialization	Scalability
Classic MoE	High	Moderate	Limited
Hierarchical MoE	High	High	Moderate
Sparse MoE (Top-K)	Low	Moderate-High	High
Soft MoE	High	High	Moderate
Hard MoE	Low	Moderate	High
Switch Transformers	Very Low	Moderate	Very High

3.7. Summary

The architectural evolution of MoE has led to various design choices, each with trade-offs in computational efficiency, specialization, and scalability [37]. While classic MoE models provide a strong foundation, recent innovations such as sparse MoE and hierarchical architectures have made MoE more practical for large-scale applications. The choice of architecture depends on the specific task requirements, computational constraints, and the need for model interpretability. In the following sections, we will discuss training methodologies, optimization techniques, and real-world applications of MoE models [38].

4. Training Methodologies and Optimization Techniques

Training a Mixture of Experts (MoE) model presents unique challenges compared to standard neural networks due to the dynamic expert selection mechanism, sparse activation, and potential imbalances in expert utilization. Effective training methodologies are crucial for ensuring stable convergence, optimal resource allocation, and improved model generalization. This section explores various training strategies, regularization techniques, and optimization improvements designed to enhance the performance of MoE models [39].

4.1. Standard Training Procedure

Training an MoE model involves optimizing both the expert networks and the gating function [40]. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, the training objective is typically to minimize a loss function of the form:

$$\mathcal{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \lambda \mathcal{R}(\theta), \tag{5}$$

 doi:10.20944/preprints202503.0611.v1

where $\ell(y_i, \hat{y}_i)$ represents the primary loss function (e.g., cross-entropy or mean squared error), and $\mathcal{R}(\theta)$ is a regularization term to encourage load balancing and prevent expert over-specialization [41]. The MoE model is trained using gradient-based optimization methods such as Stochastic Gradient Descent (SGD) or Adam [42].

4.2. Challenges in Training MoE Models

Training MoE models comes with several key challenges:

- **Expert Imbalance:** Some experts may receive significantly more data than others, leading to inefficient learning and underutilized model capacity.
- **Mode Collapse:** The gating function may learn to favor only a subset of experts, resulting in reduced diversity among experts.
- **Gradient Routing Issues:** Sparse activation of experts can lead to unstable gradients, making optimization difficult.
- **Scalability Concerns:** Large-scale MoE models require careful coordination across multiple GPUs or TPUs to avoid communication bottlenecks [43].

To mitigate these issues, various regularization and optimization techniques have been proposed.

4.3. Load Balancing and Regularization Techniques

To ensure efficient use of all experts and prevent imbalance, additional loss terms are often introduced:

4.3.1. Entropy-Based Regularization

Encouraging a more uniform distribution of gating probabilities across experts can help mitigate expert imbalance. One approach is to maximize the entropy of the gating distribution:

$$\mathcal{L}_{\text{entropy}} = -\sum_{i=1}^{K} G_i(x) \log G_i(x), \tag{6}$$

which prevents the gating function from collapsing to a few dominant experts.

4.3.2. Load Balancing Loss

Inspired by Switch Transformers [30], a load-balancing loss is often introduced to encourage even usage across experts [44]. One such formulation minimizes the variance in expert assignment:

$$\mathcal{L}_{\text{balance}} = \sum_{i=1}^{K} \left(\frac{C_i}{N} - \frac{1}{K} \right)^2, \tag{7}$$

where C_i represents the number of times expert E_i was selected, and N is the total number of training samples.

4.3.3. Noisy Gating

Noisy gating [45] introduces Gaussian noise to the gating network's logits to encourage exploration and prevent premature convergence to a small set of experts:

$$G_i(x) = \frac{\exp(W_i^T x + \epsilon_i)}{\sum_{j=1}^{K} \exp(W_j^T x + \epsilon_j)}, \tag{8}$$

where ϵ_i is drawn from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

4.4. Sparse and Efficient Training Strategies

Given that MoE models often use only a subset of experts for each input, optimizing for sparse computation is critical [46]. Several techniques address the efficiency challenges in MoE training:

 doi:10.20944/preprints202503.0611.v1

4.4.1. Top-K Expert Selection

Instead of routing an input to all experts, a common approach is to select only the top-k experts with the highest gating scores [47]. This significantly reduces computation and memory usage [48].

4.4.2. Gradient Routing and Backpropagation

Since only a subset of experts is active for each forward pass, gradient updates must be carefully managed:

- **Sparse Gradient Updates:** Only the active experts receive weight updates, reducing computational overhead [49].
- **Gradient Clipping:** Large MoE models often suffer from unstable gradients. Clipping gradient norms prevents exploding gradients [50].
- **Auxiliary Gradient Losses:** Additional losses on expert selection can encourage better gradient flow through the gating network.

4.4.3. Efficient Parallelization

Large-scale MoE models require distributed training across multiple devices [51]. Several strategies have been developed to optimize this:

- **Expert Parallelism:** Experts are distributed across different GPUs/TPUs, with each device handling only a subset of experts [52].
- **Model Parallelism:** Instead of replicating experts on every device, different devices specialize in different subsets of experts.
- **Token-Level Routing:** Instead of assigning an entire batch to a single expert, token-level routing assigns individual tokens to different experts, improving efficiency in sequence models [53].

4.5. Advanced Optimization Techniques

Several advanced techniques have been proposed to further improve MoE training:

4.5.1. Knowledge Distillation for MoE

Knowledge distillation [54] can be applied to MoE models to improve expert generalization and reduce redundancy. A distilled teacher model provides soft labels to regularize expert outputs, preventing over-specialization.

4.5.2. Reinforcement Learning-Based Gating

Some studies explore reinforcement learning (RL) for optimizing expert selection [55]. Instead of a simple softmax-based gating mechanism, the gating network is treated as an RL agent that learns to allocate inputs based on long-term rewards [56].

4.5.3. Multi-Task Learning with MoE

MoE models are particularly well-suited for multi-task learning, where different experts specialize in different tasks. Joint optimization of multiple tasks can improve expert diversity and encourage cross-task knowledge sharing.

4.6. Summary

Training MoE models effectively requires addressing challenges such as expert imbalance, sparse gradient updates, and scalability concerns [57]. Various techniques, including entropy regularization, load-balancing losses, and noisy gating, have been introduced to improve training stability [21]. Additionally, sparse computation strategies, parallelization techniques, and reinforcement learning-based optimizations have further enhanced the efficiency of MoE models. The next section explores the practical applications of MoE in various domains, highlighting its impact on large-scale machine learning.

 doi:10.20944/preprints202503.0611.v1

5. Applications of Mixture of Experts

Mixture of Experts (MoE) has found widespread applications across various domains, leveraging its ability to efficiently allocate computational resources and specialize in different data distributions [58]. From natural language processing to computer vision and scientific computing, MoE models have demonstrated significant advantages in handling large-scale and complex tasks. This section explores key application areas where MoE has been successfully employed.

5.1. Natural Language Processing

One of the most prominent applications of MoE is in natural language processing (NLP), particularly in training large-scale language models. The ability to activate only a subset of parameters per input makes MoE well-suited for efficient and scalable deep learning architectures.

5.1.1. Large Language Models

Recent advances in NLP have demonstrated that scaling up model size leads to significant improvements in performance. MoE has been widely used in training large-scale transformer-based models, such as:

- **Switch Transformers** [30]: A sparse MoE-based transformer architecture that activates only a single expert per token, achieving state-of-the-art performance while reducing computational cost [59].
- **GShard** [24]: A framework for training large-scale multilingual models using MoE, enabling efficient parallelism across distributed hardware.
- **GLaM** [21]: A scalable MoE model that dynamically routes tokens to different experts, achieving significant efficiency gains over dense transformers [60].

MoE-based language models have been deployed in various real-world applications, including machine translation, text summarization, and conversational AI.

5.1.2. Multilingual and Cross-Lingual Models

MoE has proven particularly effective in multilingual NLP tasks. By allocating different experts to different languages, MoE enables efficient parameter sharing while allowing specialization for each language [61]. This has led to improvements in cross-lingual transfer learning and low-resource language modeling.

5.2. Computer Vision

While MoE has been predominantly explored in NLP, it has also gained traction in computer vision applications. By dynamically selecting experts based on input features, MoE architectures can improve the efficiency of deep convolutional and transformer-based vision models.

5.2.1. Efficient Image Classification

MoE has been integrated into vision transformers (ViTs) to enable adaptive computation. Examples include:

- **Vision MoE** [38]: A scalable MoE-based vision transformer that achieves state-of-the-art results on ImageNet while reducing computational overhead.
- **Conditional Computation in CNNs**: MoE has been used to dynamically allocate convolutional filters based on input complexity, improving efficiency in image classification tasks [62–64].

5.2.2. Object Detection and Segmentation

In object detection and semantic segmentation, MoE models can allocate specialized experts for different object categories or regions of interest. This allows for more precise feature extraction and improved accuracy in large-scale vision datasets [65].

 doi:10.20944/preprints202503.0611.v1

5.3. Speech Processing

MoE has also been applied in speech recognition and synthesis, offering improved efficiency and adaptability [66].

5.3.1. Automatic Speech Recognition (ASR)

In ASR systems, MoE models can specialize in different phonetic variations, dialects, or background noise conditions [67]. This improves the robustness of speech recognition models in real-world scenarios.

5.3.2. Text-to-Speech (TTS)

In TTS models, MoE can be used to generate more natural and expressive speech by assigning different experts to different speech styles, accents, or emotional tones [68].

5.4. Recommendation Systems

MoE has been widely adopted in recommendation systems due to its ability to model diverse user preferences effectively.

5.4.1. Personalized Recommendations

Many large-scale recommendation engines, such as those used by e-commerce and streaming platforms, use MoE to model user-item interactions more efficiently [69]. By allocating different experts to different user groups or content types, MoE improves recommendation accuracy [70].

5.4.2. Google's YouTube and Google Play Recommendations

Google has deployed MoE-based recommendation systems in platforms like YouTube and Google Play, enabling more personalized content recommendations at scale [71].

5.5. Scientific Computing and Simulation

MoE models are increasingly being used in scientific domains, where complex simulations require adaptive computation [72].

5.5.1. Climate Modeling

In climate simulations, MoE can be used to allocate different experts to different climate regions, improving model accuracy while maintaining computational efficiency.

5.5.2. Protein Folding and Drug Discovery

MoE has been applied in bioinformatics and computational chemistry, where specialized experts can be trained on different molecular structures or biochemical interactions.

5.6. Robotics and Control Systems

In robotics and autonomous systems, MoE enables adaptive decision-making by dynamically selecting experts based on environmental conditions [73].

5.6.1. Autonomous Vehicles

MoE has been used in self-driving cars to specialize in different driving scenarios, such as highway driving, urban navigation, and off-road terrain.

5.6.2. Reinforcement Learning

In reinforcement learning (RL), MoE is used to improve exploration and exploitation strategies by dynamically selecting policies based on environmental feedback.

 doi:10.20944/preprints202503.0611.v1

5.7. Summary

MoE has demonstrated significant advantages across a wide range of applications, from large-scale language models and computer vision to recommendation systems and scientific computing. By dynamically allocating computational resources and enabling expert specialization, MoE provides a powerful framework for handling complex and diverse data distributions. The next section will discuss challenges and future directions in MoE research, focusing on scalability, interpretability, and real-world deployment considerations.

6. Challenges and Future Directions

Despite the significant success of Mixture of Experts (MoE) models in various domains, several challenges remain that hinder their widespread adoption and efficient deployment [74]. These challenges include issues related to scalability, interpretability, fairness, and hardware optimization. In this section, we explore these limitations and discuss potential research directions to overcome them [75].

6.1. Scalability and Computational Efficiency

One of the primary motivations for MoE is its ability to scale deep learning models efficiently by activating only a subset of parameters for each input [76]. However, as MoE models continue to grow in size, new scalability challenges emerge.

6.1.1. Memory and Communication Overhead

MoE models often require multiple experts to be distributed across different GPUs or TPUs. This introduces additional memory constraints and inter-device communication overhead. The need for frequent synchronization among experts can lead to bottlenecks, limiting the efficiency of distributed training [77]. **Potential Solutions:**

- **Sparse Communication Strategies:** Reducing inter-GPU communication by updating only a subset of experts per step [78].
- **Efficient Parameter Sharing:** Using techniques like expert merging or weight tying to reduce memory footprint [79].
- **Hierarchical MoE Architectures:** Introducing multi-level expert selection to balance computational cost [80].

6.1.2. Training Stability and Convergence

MoE models often suffer from instability during training due to sparse expert selection and imbalanced expert utilization [81]. The gating mechanism may cause some experts to receive significantly more updates than others, leading to inefficient learning [82]. **Potential Solutions:**

- **Improved Load Balancing Mechanisms:** Advanced regularization techniques to encourage even usage of experts.
- **Adaptive Expert Routing:** Dynamically adjusting expert selection criteria based on training progress [83].
- **Gradient Clipping and Normalization:** Preventing extreme weight updates that cause instability.

6.2. Interpretability and Explainability

A major concern with MoE models is their lack of interpretability. Since decisions are made dynamically by the gating network, understanding why certain experts are chosen for specific inputs remains challenging.

6.2.1. Understanding Expert Behavior

Unlike traditional deep learning models where all parameters contribute to every prediction, MoE models selectively activate experts. This raises questions about whether experts learn meaningful and distinct representations. **Potential Solutions:**

 doi:10.20944/preprints202503.0611.v1

- **Visualization Techniques:** Using attention maps or activation patterns to understand expert specialization.
- **Explainable AI (XAI) Methods:** Applying techniques like SHAP values to analyze gating decisions.
- **Explicit Expert Constraints:** Encouraging experts to learn distinct, non-overlapping feature representations.

6.3. Fairness and Bias in Expert Selection

MoE models, particularly in applications like recommendation systems and hiring algorithms, can inherit and amplify biases in data. If certain experts become dominant for specific subgroups, it may lead to biased or unfair outcomes [84]. **Potential Solutions:**

- **Diversity Regularization:** Enforcing constraints that encourage diverse expert activation across different demographic groups.
- **Bias Detection in Expert Assignments:** Analyzing whether specific user groups consistently get routed to the same experts.
- **Fairness-Aware MoE Models:** Designing MoE architectures that explicitly account for fairness objectives [85].

6.4. Robustness and Generalization

While MoE models excel in many large-scale tasks, they often struggle with robustness, especially in out-of-distribution (OOD) scenarios.

6.4.1. Generalization to Unseen Data

Since experts are trained on subsets of data, they may overfit to specific distributions and fail to generalize to novel inputs. **Potential Solutions:**

- **Meta-Learning for MoE:** Training experts to generalize across a broader range of tasks [86].
- **Uncertainty-Aware Gating Mechanisms:** Introducing probabilistic gating models to handle unseen scenarios more effectively [87].
- **Data Augmentation for Expert Diversity:** Ensuring each expert is exposed to a more diverse set of inputs [88].

6.5. Hardware and Deployment Challenges

Deploying MoE models in real-world applications poses unique challenges, particularly in environments with limited computational resources.

6.5.1. Efficient Inference

While MoE reduces the number of active parameters per inference step, it still introduces overhead due to the dynamic routing mechanism. **Potential Solutions:**

- **Edge-Friendly MoE Architectures:** Developing lightweight MoE variants optimized for edge and mobile devices [64].
- **Distilled MoE Models:** Using knowledge distillation to create smaller, more efficient models without sacrificing performance.
- **Latency-Aware Expert Selection:** Adapting the gating mechanism to prioritize faster experts under real-time constraints [89].

6.6. Future Research Directions

Given the challenges outlined above, several promising research directions can help advance MoE models:

- **Self-Supervised and Unsupervised MoE:** Exploring how MoE models can be effectively trained without labeled data [90].

 doi:10.20944/preprints202503.0611.v1

- **Neuroscientific Inspiration for MoE Design:** Drawing insights from biological neural networks to develop more efficient and adaptive expert selection mechanisms.
- **Hybrid MoE Architectures:** Combining MoE with other architectural paradigms, such as reinforcement learning or meta-learning, to improve adaptability [20].
- **Theoretical Understanding of MoE:** Developing a deeper mathematical foundation to explain MoE's efficiency and performance [91].

6.7. Summary

While MoE models offer significant advantages in scalability and efficiency, they also introduce new challenges related to training stability, interpretability, fairness, and hardware optimization [92]. Addressing these issues will require advances in algorithm design, regularization techniques, and hardware-aware implementations [93]. Future research should focus on improving expert utilization, ensuring fairness, and making MoE models more robust and interpretable. The next section concludes this survey by summarizing key insights and highlighting the broader implications of MoE in machine learning.

7. Conclusionx

Mixture of Experts (MoE) has emerged as a powerful paradigm for scaling deep learning models efficiently while maintaining computational feasibility [94]. By selectively activating subsets of parameters, MoE enables specialized learning, leading to improved performance in various domains such as natural language processing, computer vision, speech recognition, recommendation systems, and scientific computing. This survey has provided a comprehensive overview of MoE, covering its fundamental concepts, training mechanisms, architectural variations, key applications, challenges, and future research directions [95].

7.1. Key Insights

Through this survey, several key insights about MoE have been highlighted:

- **Scalability and Efficiency:** MoE models allow the training and deployment of large-scale models with billions of parameters while maintaining manageable computational costs through sparse activation [96].
- **Domain-Specific Advantages:** MoE has demonstrated remarkable success in NLP, particularly in large-scale language models, as well as in computer vision, recommendation systems, and scientific simulations.
- **Challenges in Training and Deployment:** Despite its advantages, MoE introduces issues related to training stability, expert imbalance, and increased memory and communication overhead.
- **Interpretability and Fairness Considerations:** Understanding expert behavior and ensuring fairness in expert routing are critical concerns for deploying MoE in real-world applications [97].
- **Future Research Directions:** Promising areas for future work include improving load balancing, developing fairness-aware MoE architectures, exploring unsupervised and self-supervised MoE, and optimizing MoE for real-time inference and deployment on edge devices [98].

7.2. Broader Implications

The success of MoE in recent years has far-reaching implications for the future of artificial intelligence. The ability to scale models efficiently while maintaining specialization offers new opportunities for advancing AI capabilities across multiple disciplines [99]. However, the challenges associated with MoE also underscore the need for further research in areas such as fairness, robustness, and hardware optimization. As AI systems continue to grow in scale and complexity, MoE is likely to play a central role in shaping the next generation of deep learning architectures [100–102].

 doi:10.20944/preprints202503.0611.v1

7.3. Final Remarks

MoE represents a significant step forward in designing more efficient and scalable AI models. While many challenges remain, ongoing research and technological advancements will likely refine MoE architectures, making them more accessible and practical for widespread adoption. As the field continues to evolve, addressing the limitations of MoE while capitalizing on its strengths will be crucial for unlocking its full potential in machine learning and beyond.

References

1. Rosenbaum, C.; Klinger, T.; Riemer, M. Routing networks: Adaptive selection of non-linear functions for multi-task learning. *arXiv preprint arXiv:1711.01239* **2017**.
2. Chi, Z.; Dong, L.; Huang, S.; Dai, D.; Ma, S.; Patra, B.; Singhal, S.; Bajaj, P.; Song, X.; Mao, X.L.; et al. On the representation collapse of sparse mixture of experts. *Advances in Neural Information Processing Systems* **2022**, *35*, 34600–34613.
3. 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.
4. Team, Q. Introducing Qwen1.5, 2024.
5. 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.
6. Chen, C.; Li, M.; Wu, Z.; Yu, D.; Yang, C. Ta-moe: Topology-aware large scale mixture-of-expert training. *Advances in Neural Information Processing Systems* **2022**, *35*, 22173–22186.
7. Gale, T.; Narayanan, D.; Young, C.; Zaharia, M. Megablocks: Efficient sparse training with mixture-of-experts. *Proceedings of Machine Learning and Systems* **2023**, *5*.
8. Muqeeth, M.; Liu, H.; Raffel, C. Soft merging of experts with adaptive routing. *arXiv preprint arXiv:2306.03745* **2023**.
9. 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**.
10. Roller, S.; Sukhbaatar, S.; Weston, J.; et al. Hash layers for large sparse models. *Advances in Neural Information Processing Systems* **2021**, *34*, 17555–17566.
11. Jacobs, R.A.; Jordan, M.I.; Nowlan, S.J.; Hinton, G.E. Adaptive mixtures of local experts. *Neural computation* **1991**, *3*, 79–87.
12. 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.
13. 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*.
14. Zhang, X.; Shen, Y.; Huang, Z.; Zhou, J.; Rong, W.; Xiong, Z. Mixture of Attention Heads: Selecting Attention Heads Per Token. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 4150–4162.
15. Kim, Y.J.; Awan, A.A.; Muzio, A.; Salinas, A.F.C.; Lu, L.; Hendy, A.; Rajbhandari, S.; He, Y.; Awadalla, H.H. Scalable and efficient moe training for multitask multilingual models. *arXiv preprint arXiv:2109.10465* **2021**.
16. 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**.
17. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* **2017**, *114*, 3521–3526.
18. 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.
19. 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*.
20. Lample, G.; Sablayrolles, A.; Ranzato, M.; Denoyer, L.; Jégou, H. Large memory layers with product keys. *Advances in Neural Information Processing Systems* **2019**, *32*.

 doi:10.20944/preprints202503.0611.v1

21. 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.
22. Du, Y.; Zhao, S.; Zhao, D.; Ma, M.; Chen, Y.; Huo, L.; Yang, Q.; Xu, D.; Qin, B. MoGU: A Framework for Enhancing Safety of Open-Sourced LLMs While Preserving Their Usability. *arXiv preprint arXiv:2405.14488* **2024**.
23. Shen, Y.; Zhang, Z.; Cao, T.; Tan, S.; Chen, Z.; Gan, C. Moduleformer: Learning modular large language models from uncurated data. *arXiv preprint arXiv:2306.04640* **2023**.
24. 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**.
25. 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**.
26. 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.
27. Xue, F.; Shi, Z.; Wei, F.; Lou, Y.; Liu, Y.; You, Y. Go wider instead of deeper. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2022, Vol. 36, pp. 8779–8787.
28. 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.
29. Dua, D.; Bhosale, S.; Goswami, V.; Cross, J.; Lewis, M.; Fan, A. Tricks for Training Sparse Translation Models. 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. 3340–3345.
30. 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.
31. He, X.O. Mixture of A Million Experts. *arXiv preprint arXiv:2407.04153* **2024**.
32. Zheng, L.; Li, Z.; Zhang, H.; Zhuang, Y.; Chen, Z.; Huang, Y.; Wang, Y.; Xu, Y.; Zhuo, D.; Xing, E.P.; et al. Alpa: Automating inter-and {Intra-Operator} parallelism for distributed deep learning. In Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22), 2022, pp. 559–578.
33. Fedus, W.; Dean, J.; Zoph, B. A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667* **2022**.
34. 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.
35. 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.
36. Shazeer, N. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202* **2020**.
37. Jiang, A.Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D.S.; Casas, D.d.l.; Hanna, E.B.; Bressand, F.; et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088* **2024**.
38. 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.
39. 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**.
40. Zhang, Z.; Liu, S.; Yu, J.; Cai, Q.; Zhao, X.; Zhang, C.; Liu, Z.; Liu, Q.; Zhao, H.; Hu, L.; et al. M3oE: Multi-Domain Multi-Task Mixture-of Experts Recommendation Framework. In Proceedings of the Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 893–902.
41. Zhai, M.; He, J.; Ma, Z.; Zong, Z.; Zhang, R.; Zhai, J. {SmartMoE}: Efficiently Training {Sparsely-Activated} Models through Combining Offline and Online Parallelization. In Proceedings of the 2023 USENIX Annual Technical Conference (USENIX ATC 23), 2023, pp. 961–975.

42. Antoniak, S.; Jaszczur, S.; Krutul, M.; Pióro, M.; Krajewski, J.; Ludziejewski, J.; Odrzygóźdź, T.; Cygan, M. Mixture of Tokens: Efficient LLMs through Cross-Example Aggregation. *arXiv preprint arXiv:2310.15961* **2023**.
43. 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.
44. He, S.; Fan, R.Z.; Ding, L.; Shen, L.; Zhou, T.; Tao, D. Merging Experts into One: Improving Computational Efficiency of Mixture of Experts. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 14685–14691.
45. 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**.
46. Jacobs, S.A.; Tanaka, M.; Zhang, C.; Zhang, M.; Song, L.; Rajbhandari, S.; He, Y. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509* **2023**.
47. Xiao, D.; Zhang, H.; Li, Y.; Sun, Y.; Tian, H.; Wu, H.; Wang, H. ERNIE-GEN: an enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In Proceedings of the Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 3997–4003.
48. Ren, X.; Zhou, P.; Meng, X.; Huang, X.; Wang, Y.; Wang, W.; Li, P.; Zhang, X.; Podolskiy, A.; Arshinov, G.; et al. Pangu-Σ: Towards trillion parameter language model with sparse heterogeneous computing. *arXiv preprint arXiv:2303.10845* **2023**.
49. Lieber, O.; Lenz, B.; Bata, H.; Cohen, G.; Osin, J.; Dalmedigos, I.; Safahi, E.; Meirom, S.; Belinkov, Y.; Shalev-Shwartz, S.; et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887* **2024**.
50. 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.
51. 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.
52. 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.
53. Dai, D.; Deng, C.; Zhao, C.; Xu, R.; Gao, H.; Chen, D.; Li, J.; Zeng, W.; Yu, X.; Wu, Y.; et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066* **2024**.
54. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* **2015**.
55. 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.
56. Xue, F.; Zheng, Z.; Fu, Y.; Ni, J.; Zheng, Z.; Zhou, W.; You, Y. Openmoe: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2402.01739* **2024**.
57. Clark, A.; de Las Casas, D.; Guy, A.; Mensch, A.; Paganini, M.; Hoffmann, J.; Damoc, B.; Hechtman, B.; Cai, T.; Borgeaud, S.; et al. Unified scaling laws for routed language models. In Proceedings of the International conference on machine learning. PMLR, 2022, pp. 4057–4086.
58. 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**.
59. 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**.
60. Chen, T.; Zhang, Z.; JAISWAL, A.K.; Liu, S.; Wang, Z. Sparse MoE as the New Dropout: Scaling Dense and Self-Slimmable Transformers. In Proceedings of the The Eleventh International Conference on Learning Representations, 2022.

 doi:10.20944/preprints202503.0611.v1

61. 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.
62. 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**.
63. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
64. 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**.
65. Bi, X.; Chen, D.; Chen, G.; Chen, S.; Dai, D.; Deng, C.; Ding, H.; Dong, K.; Du, Q.; Fu, Z.; et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954* **2024**.
66. 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**.
67. 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**.
68. Artetxe, M.; Bhosale, S.; Goyal, N.; Mihaylov, T.; Ott, M.; Shleifer, S.; Lin, X.V.; Du, J.; Iyer, S.; Pasunuru, R.; et al. Efficient large scale language modeling with mixtures of experts. *arXiv preprint arXiv:2112.10684* **2021**.
69. Ainslie, J.; Lee-Thorp, J.; de Jong, M.; Zemlyanskiy, Y.; Lebron, F.; Sanghai, S. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 4895–4901.
70. Dai, D.; Dong, L.; Ma, S.; Zheng, B.; Sui, Z.; Chang, B.; Wei, F. StableMoE: Stable Routing Strategy for Mixture of Experts. In Proceedings of the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 7085–7095.
71. 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**.
72. Rajbhandari, S.; Ruwase, O.; Rasley, J.; Smith, S.; He, Y. Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. In Proceedings of the Proceedings of the international conference for high performance computing, networking, storage and analysis, 2021, pp. 1–14.
73. 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.
74. Shen, Y.; Guo, Z.; Cai, T.; Qin, Z. JetMoE: Reaching Llama2 Performance with 0.1 M Dollars. *arXiv preprint arXiv:2404.07413* **2024**.
75. Zeng, Z.; Miao, Y.; Gao, H.; Zhang, H.; Deng, Z. AdaMoE: Token-Adaptive Routing with Null Experts for Mixture-of-Experts Language Models. *arXiv preprint arXiv:2406.13233* **2024**.
76. 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.
77. Team, L.M. LLaMA-MoE: Building Mixture-of-Experts from LLaMA with Continual Pre-training, 2023.
78. 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**.
79. Costa-jussà, M.R.; Cross, J.; Çelebi, O.; Elbayad, M.; Heafield, K.; Heffernan, K.; Kalbassi, E.; Lam, J.; Licht, D.; Maillard, J.; et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672* **2022**.
80. 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**.
81. Shi, S.; Pan, X.; Wang, Q.; Liu, C.; Ren, X.; Hu, Z.; Yang, Y.; Li, B.; Chu, X. ScheMoE: An Extensible Mixture-of-Experts Distributed Training System with Tasks Scheduling. In Proceedings of the Proceedings of the Nineteenth European Conference on Computer Systems, 2024, pp. 236–249.
82. 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.

 doi:10.20944/preprints202503.0611.v1

83. 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.
84. Dou, S.; Zhou, E.; Liu, Y.; Gao, S.; Zhao, J.; Shen, W.; Zhou, Y.; Xi, Z.; Wang, X.; Fan, X.; et al. The Art of Balancing: Revolutionizing Mixture of Experts for Maintaining World Knowledge in Language Model Alignment. *arXiv preprint arXiv:2312.09979* **2023**.
85. 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.
86. Zhang, Q.; Zou, B.; An, R.; Liu, J.; Zhang, S. MoSA: Mixture of Sparse Adapters for Visual Efficient Tuning. *arXiv preprint arXiv:2312.02923* **2023**.
87. 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.
88. Zhang, Z.; Lin, Y.; Liu, Z.; Li, P.; Sun, M.; Zhou, J. MoEfication: Transformer Feed-forward Layers are Mixtures of Experts. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, 2022, pp. 877–890.
89. 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**.
90. 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.
91. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* **2022**, *35*, 27730–27744.
92. 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**.
93. Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; Auli, M. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038* **2019**.
94. 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**.
95. 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**.
96. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
97. 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.
98. 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**.
99. 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.
100. Zhang, Y.; Cai, R.; Chen, T.; Zhang, G.; Zhang, H.; Chen, P.Y.; Chang, S.; Wang, Z.; Liu, S. Robust Mixture-of-Expert Training for Convolutional Neural Networks. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 90–101.

 doi:10.20944/preprints202503.0611.v1

101. Li, Y.; Jiang, S.; Hu, B.; Wang, L.; Zhong, W.; Luo, W.; Ma, L.; Zhang, M. Uni-MoE: Scaling Unified Multimodal LLMs with Mixture of Experts. *arXiv preprint arXiv:2405.11273* **2024**.
102. Zhang, R.; Han, J.; Liu, C.; Zhou, A.; Lu, P.; Li, H.; Gao, P.; Qiao, Y. LLaMA-Adapter: Efficient Fine-tuning of Large Language Models with Zero-initialized Attention. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.