THE EVOLUTION OF MIXTURE OF EXPERTS: A SURVEY FROM BASICS TO BREAKTHROUGHS

Arpita Vats[1]*, Rahul Raja*[2]*, Vinija Jain*[3]*, Aman Chadha*[3]

[1]Santa Clara University, CA
[2]Carnegie Mellon University, PA
[3]Stanford University, CA

ABSTRACT

The Mixture of Experts (MoE) architecture has evolved as a powerful and versatile approach for improving the performance and efficiency of deep learning models. This survey aims to provide the fundamental principles of MoE in details, a paradigm that harnesses the collective power of multiple specialized "expert" models working in concert to tackle complex tasks.Our exploration is a thorough analysis of the core components that constitute the MoE framework. We begin by dissecting the routing mechanism, a crucial element responsible for dynamically assigning input data to the most appropriate expert models. This routing process is pivotal in ensuring that each expert's specialized knowledge is optimally utilized. The survey places significant emphasis on expert specialization, a key feature that sets MoE apart from traditional architectures. We examine various strategies for developing and training specialized experts, exploring how this specialization enables MoE models to effectively handle diverse and multifaceted problems.Load balancing, another critical aspect of MoE systems, receives thorough attention. We discuss techniques for efficiently distributing computational resources among experts, ensuring optimal model performance while managing hardware constraints. This section provides insights into the delicate balance between model capacity and computational efficiency. Furthermore, we provide an in-depth discussion of the expert models themselves, examining their architectural designs, training methodologies, and how they interact within the larger MoE framework. This includes an analysis of different types of experts, from simple neural networks to more complex, task-specific architectures. The survey also navigates through diverse research avenues within the MoE landscape, highlighting recent advancements and innovative applications across various domains of machine learning. We pay particular attention to the burgeoning use of MoE in two rapidly evolving fields: computer vision and large language model (LLM) scaling. By providing this comprehensive overview, our survey aims to offer researchers and practitioners a deep understanding of MoE's capabilities, current applications, and potential future directions in the ever-evolving landscape of deep learning.

1. INTRODUCTION

The MoE concept is a type of ensemble learning technique initially developed within the field of artificial neural networks. It introduces the idea of training experts on specific subtasks of a complex predictive modeling problem. In a typical ensemble scenario, all models are trained on the same dataset, and their outputs are combined through simple averaging, weighted mean, or majority voting. However, in an MoE architecture, each "expert" model within the ensemble is only trained on a subset of data where it can achieve optimal performance, thus narrowing the model's focus. Put simply, MoE is an architecture that divides input data into multiple sub-tasks and trains a group of experts to specialize in each sub-task. These experts can be thought of as smaller, specialized models that are better at solving their respective sub-tasks. The popularity of MoE only rose recently as the appearance of Large Language Models (LLMs) and transformer-based models in general swept through the machine learning field. Consequently, this is because of modern datasets' increased complexity and size. Each dataset contains different regimes with vastly different relationships between the features and the labels. The Mixture of Experts (MoE) by Jacob *et al*. [1] concept is a type of ensemble learning technique initially developed within the field of artificial neural networks. It introduces the idea of training experts on specific subtasks of a complex predictive modeling problem. In a typical ensemble scenario, all models are trained on the same dataset, and their outputs are combined through simple averaging, weighted mean, or majority voting. However, in MoE, each "expert" model within the ensemble is only trained on a subset of data where it can achieve optimal performance, thus narrowing the model's focus. Put simply, MoE is an architecture that divides input data into multiple sub-tasks and trains a group of experts to specialize in each sub-task. These experts can be thought of as smaller, specialized models that are better at solving their respective sub-tasks. To appreciate the essence of MoE, it is crucial to understand its architectural elements:

- **Division of dataset into local subsets**: First, the predictive modeling problem is divided into subtasks. This division often requires domain knowledge or employs an unsupervised clustering algorithm. It's important to clarify that clustering is not based on the feature vectors' similarities. Instead, it's executed based on the correlation among the relationships that the features share with the labels.
- **Expert models**: These are the specialized neural network layers or experts that are trained to excel at specific sub-tasks. Each expert receives the same input pattern and processes it according to its specialization. An expert is trained for each subset of the data. Typically, the experts themselves can be any model, from Support Vector Machines (SVM) to neural networks. Each expert model receives the same input pattern and makes a prediction.
- **Gating network (Router)**: The gating network, also called the router, is responsible for selecting which experts to use for each input data. It works by estimating the compatibility between the input data and each expert, and then outputs a softmax distribution over the experts. This distribution is used as the weights to combine the outputs of the expert layers.

This model helps interpret predictions made by each expert and decide which expert to trust for a given input.

- **Pooling method**: Finally, an aggregation mechanism is needed to make a prediction based on the output from the gating network and the experts. The gating network and expert layers are jointly trained to minimize the overall loss function of the MoE model. The gating network learns to route each input to the most relevant expert layer(s), while the expert layers specialize in their assigned sub-tasks.

This divide-and-conquer approach [2] effectively delegates complex tasks to experts, enabling efficient processing and improved accuracy. Together, these components ensure that the right expert handles the right task. The gating network effectively routes each input to the most appropriate expert(s), while the experts focus on their specific areas of strength. This collaborative approach leads to a more versatile and capable overall model. The gating network and expert layers are jointly trained to minimize the overall loss function of the MoE model. The gating network learns to route each input to the most relevant expert layer(s), while the expert layers specialize in their assigned sub-tasks.This divide-and-conquer approach effectively delegates complex tasks to experts, enabling efficient processing and improved accuracy. Together, these components ensure that the right expert handles the right task. The gating network effectively routes each input to the most appropriate expert(s), while the experts focus on their specific areas of strength. This collaborative approach leads to a more versatile and capable overall model. In summary, MoEs improve efficiency by dynamically selecting a subset of model parameters (experts) for each input. This architecture allows for larger models while keeping computational costs manageable by activating only a few experts per input.

1.1. Gate Functionality

This section seeks to answer how the gating network (also called gate, router, or switch) in MoE models works under the hood. Let's explore two distinct but interconnected functions of the gate in a MoE model:

- Clustering the Data: In the context of an MoE model, clustering the data means that the gate is learning to identify and group together similar data points. There is no clustering in the traditional unsupervised learning, where the algorithm discovers clusters without any external labels. The gate learns from the training process to identify features or patterns in the data that indicate which data points are comparable to one another and ought to be handled accordingly. This is a important stage since it establishes the structure and interpretation of the data by the model.
- Mapping Experts to Clusters: when the gate has identified clusters within the data, the next part is to assign or map each cluster to the most appropriate expert within the MoE model. Each expert in the model is specialized to handle different types of data or different aspects of the problem. The gate's function direct each data point (or each group of similar data points) to the expert that is best suited to process it. The mapping is dynamic and is based on the strengths and specialties of each expert when they evolve during the training process.

In summary, the gate in an MoE model is responsible for organizing the incoming data into meaningful groups (clustering) and then efficiently allocating these groups to the most relevant expert models within the MoE system for further processing. This dual role of the gate is critical for the overall performance and efficiency of the MoE model, enabling it to handle complex tasks by leveraging the specialized skills of its various expert components.

1.2. Sparsely Gated

In 2017, an extension of the MoE paradigm suited for deep learning was proposed by Noam Shazeer *et al*. [3]. In most deep learning models, increasing model capacity generally translates to improved performance when datasets are sufficiently large. Generally, when the entire model is activated by every example, it can lead to "a roughly quadratic blow-up in training costs, as both the model size and the number of training examples increase", stated by Shazeer *et al*. Although the disadvantages of dense models are clear, there have been various challenges for an effective conditional computation method targeted toward modern deep learning models, mainly for the following reasons:

- Modern computing devices like GPUs and TPUs perform better in arithmetic operations than in network branching.
- Larger batch sizes benefit performance but are reduced by conditional computation.
- Network bandwidth can limit computational efficiency, notably affecting embedding layers.
- Some schemes might need loss terms to attain required sparsity levels, impacting model quality and load balance.
- Model capacity is vital for handling vast data sets, a challenge that current conditional computation literature doesn't adequately address.

The MoE technique presented by Shazeer *et al*. aims to achieve conditional computation while addressing the abovementioned issues. They could increase model capacity by more than a thousandfold while only sustaining minor computational efficiency losses. The authors introduced a new type of network layer called the "Sparsely-Gated MoE Layer." They are built on previous iterations of MoE and aim to provide a general-purpose neural network component that can be adapted to different types of tasks. The Sparsely-Gated MoE architecture (henceforth, referred to as the MoE architecture), consists of numerous expert networks, each being a simple feed-forward neural network and a trainable gating network. The gating network is responsible for selecting a sparse combination of these experts to process each input. The fascinating feature here is the use of sparsity in the gating function. This means that for every input instance, the gating network only selects a few experts for processing, keeping the rest inactive. This sparsity and expert selection is achieved dynamically for each input, making the entire process highly flexible and adaptive. Notably, the computational efficiency is preserved since inactive parts of the network are not processed. The MoE layer can be stacked hierarchically, where the primary MoE selects a sparsely weighted combination of "experts." Each combination utilizes a MoE layer. Moreover, the authors also introduced an innovative technique called Noisy Top-K Gating. This mechanism adds a tunable Gaussian noise to the gating function, retains only the top k values, and assigns the rest to negative infinity, translating to a zero gating value. Such an approach ensures the sparsity of the gating network while maintaining robustness against potential discontinuities in the gating function output. Interestingly, it also aids in load balancing across the expert networks. In their framework, both the gating network and the experts are trained jointly via back-propagation, the standard training mechanism for neural networks. The output from the gating network is a sparse, n-dimensional vector, which serves as the gate values for the n-expert networks. The

output from each expert is then weighted by the corresponding gating value to produce the final model output. The Sparse MoE architecture has been a game-changer in LLMs, allowing us to scale up modeling capacity with almost constant computational complexity, resulting breakthroughs such as the Switch Transformer, GPT-4, Mixtral-8x7b, and more.

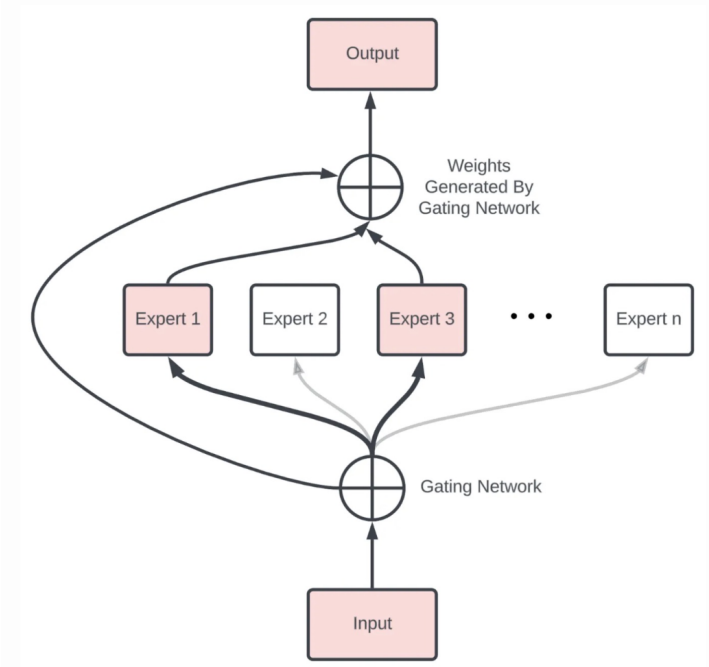

Fig. 1. Sparse Mixture of Experts Mixtral 8x22B

2. RELATED WORKS

The neural network's information absorption capacity is constrained by its parameter count. Conditional computation, a concept theorized to significantly boost model capacity without proportional computation increases, faces practical challenges. Traditional static neural network architectures apply uniform functions to all examples, whereas input-dependent models customize functions for each example. While specifying a static architecture manually is easy, defining input-dependent functions for every example manually is impractical. Consequently, these functions must be inferred automatically by the model, adding complexity to optimization. To address the need for automatic architecture inference, a common approach is to construct a single large model (supernetwork) with numerous sub-networks (experts) and route examples through this network. A Sparsely-Gated Mixture-of-Experts layer (MoE) contains thousands of feed-forward sub-networks, with a trainable gating network determining a sparse combination of these experts for each example. This per-example routing mechanism processes different examples through distinct subcomponents or experts within the larger model, known as a supernetwork. This setup intuitively suggests that similar examples may traverse similar paths, while dissimilar ones might take different paths. Furthermore, example-dependent routing promotes expert specialization, where experts focus their representational capacity on transforming specific subsets of examples. The proposed MoE layer takes as an input a token representation `x` and then routes this to the best determined `top-k` experts, selected from a set $\{E_i(x)\}_{i=1}^{N}$ of N experts. The router variable W_r produces logits $h(x) = W_r \cdot x$ which are normalized via a softmax distribution over the available N experts at that layer. The gate-value for expert `i` is given by:

$$p_i(x) = \frac{e^{h(x)_i}}{\sum_{j=1}^{N} e^{h(x)_j}}$$

The `top-k` gate values are selected for routing the token `x` If τ is the set of selected `top-k` indices then the output computation of the layer is the linearly weighted combination of each expert's computation on the token by the gate value, $y = \sum_{i \in \tau} p_i(x) E_i(x)$ They apply the MoE to the tasks of language modeling and machine translation, where model capacity is critical for absorbing the vast quantities of knowledge available in the training corpora.They present model architectures in which a MoE with up to 137 billion parameters is applied convolutionally between stacked LSTM layers. On large language modeling and machine translation benchmarks, these models achieve significantly better results than state-of-the-art at lower computational cost. Mixtures of Experts combine the outputs of several "expert" networks, each of which specializes in a different part of the input space. This is achieved by training a "gating" network that maps each input to a distribution over the experts. Such models show promise for building larger networks that are still cheap to compute at test time, and more parallelizable at training time. This paper by Eigen *et al*. [4] from Google and NYU Courant in 2013 extends the Mixture of Experts to a stacked model, the Deep Mixture of Experts, with multiple sets of gating and experts. This exponentially increases the number of effective experts by associating each input with a combination of experts at each layer, yet maintains a modest model size. On a randomly translated version of the MNIST dataset, they find that the Deep Mixture of Experts automatically learns to develop location-dependent ("where") experts at the first layer, and class-specific ("what") experts at the second layer. In addition, they see that the different combinations are in use when the model is applied to a dataset of speech monophones. These demonstrate effective use of all expert combinations.

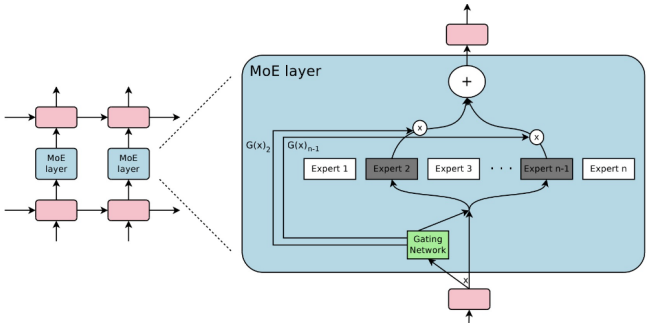

Fig. 2. This image illustrates a MoE layer embedded within a recurrent language model [3]. In this case, the sparse gating function selects two experts to perform computations. Their outputs are modulated by the outputs of the gating network

In 2017, an extension of the MoE paradigm suited for deep learning was proposed by Shazeer *et al*. [3] In most deep learning models, increasing model capacity generally translates to improved performance when datasets are sufficiently large. Generally, when the entire model is activated by every example, it can lead to "a roughly quadratic blow-up in training costs, as both the model size and the number of training examples increase". Although the disadvantages of dense models are clear, there have been various

challenges for an effective conditional computation method targeted toward modern deep learning models, mainly for the following reasons:

- Modern computing devices like GPUs and TPUs perform better in arithmetic operations than in network branching.
- Larger batch sizes benefit performance but are reduced by conditional computation.
- Network bandwidth can limit computational efficiency, notably affecting embedding layers.
- Some schemes might need loss terms to attain required sparsity levels, impacting model quality and load balance.
- Model capacity is vital for handling vast data sets, a challenge that current conditional computation literature doesn't adequately address.

The MoE technique presented by [3] aims to achieve conditional computation while addressing the abovementioned issues. They could increase model capacity by more than a thousandfold while only sustaining minor computational efficiency losses. The authors introduced a new type of network layer called the "Sparsely-Gated MoE Layer." They are built on previous iterations of MoE and aim to provide a general-purpose neural network component that can be adapted to different types of tasks. The Sparsely-Gated MoE architecture (henceforth, referred to as the MoE architecture), consists of numerous expert networks, each being a simple feed-forward neural network and a trainable gating network. The gating network is responsible for selecting a sparse combination of these experts to process each input.

2.1. Expert Choice Routing

In MoEs, expert capacity and capacity factor are crucial for ensuring balanced load distribution and efficient utilization of the model's experts. Expert capacity sets the maximum number of tokens or inputs each expert can process, while the capacity factor adjusts this capacity to provide flexibility in managing computational resources.

- Expert Capacity: C_i is the Capacity of the i^{th} expert, representing the maximum number of tokens it can handle.
- Capacity Factor: α Capacity factor, a scalar that adjusts the effective capacity of each expert.
- Gating Network: G_{ij} Gating probability or decision for the j^{th} token to be routed to the i^{th} , It can be binary (0 or 1) or a continuous value between 0 and 1.

In the context of MoEs, expert capacity and the capacity factor play critical roles in managing load distribution and expert utilization. By incorporating the capacity factor into the effective capacity constraints and penalty terms in the loss function, MoE models can achieve balanced load distribution and improved performance. The equations provided help formalize the management of expert capacities, ensuring that the model operates efficiently without overloading any single expert.

2.2. Load Balancing

Load balancing is a critical issue in MoEs, ensuring that all experts are used evenly. Without proper load balancing, some experts might be over-utilized while others are under-utilized, leading to inefficiencies and degraded model performance. Effective load balancing ensures that the computational resources are fully utilized, which enhances the model's overall effectiveness and efficiency.

1. **Loss Function Component**: The loss function in MoEs typically includes a term to encourage load balancing. This term penalizes the model when the load is unevenly distributed across the experts. The loss function can be expressed as:

$$L = L_{\text{task}} + \lambda \, L_{\text{load\_balancing}}$$

   (a) L_{task}: The primary task-specific loss (e.g., cross-entropy loss for classification).

   (b) $\lambda \, L_{\text{load\_balancing}}$: A penalty term to ensure experts are used evenly.

   (c) λ: A hyperparameter to control the importance of the load balancing term.

2. One common approach is to use the entropy of the expert selection probabilities to encourage a uniform distribution:

$$L_{load\_balancing} = -\sum_{i=1}^{N} p_i \log p_i$$

   (a) where p_i is the probability of selecting the i^{th} expert, and N is the total number of experts

Potential Solutions for Load Balancing

- Several strategies can be employed to address load balancing in MoEs:

  1. **Regularization Terms in Loss Function**
     - Include terms in the loss function that penalize uneven expert utilization.
     - Use entropy-based regularization to encourage a uniform distribution of expert usage.
  2. **Gating Networks**
     - Use a sophisticated gating mechanism to ensure more balanced expert selection.
     - Implement gating networks that consider the historical usage of experts to avoid over-reliance on specific experts.
  3. **Expert Capacity Constraints**
     - Set a maximum capacity for each expert, limiting the number of inputs an expert can handle
     - Dynamically adjust the capacity based on the current load to distribute the work more evenly.
  4. **Routing Strategies**
     - Employ advanced routing strategies that distribute inputs more evenly among experts.
     - Use probabilistic or learned routing to decide which experts should handle which inputs.
  5. **MegaBlocks Approach**
     - Proposed in MegaBlocks: Efficient Sparse Training with Mixture-of-Experts, MegaBlocks introduces block-wise parallelism and a structured sparse approach to balance the load.
     - It partitions the model into blocks and uses efficient algorithms to ensure even distribution of computational load across these blocks.

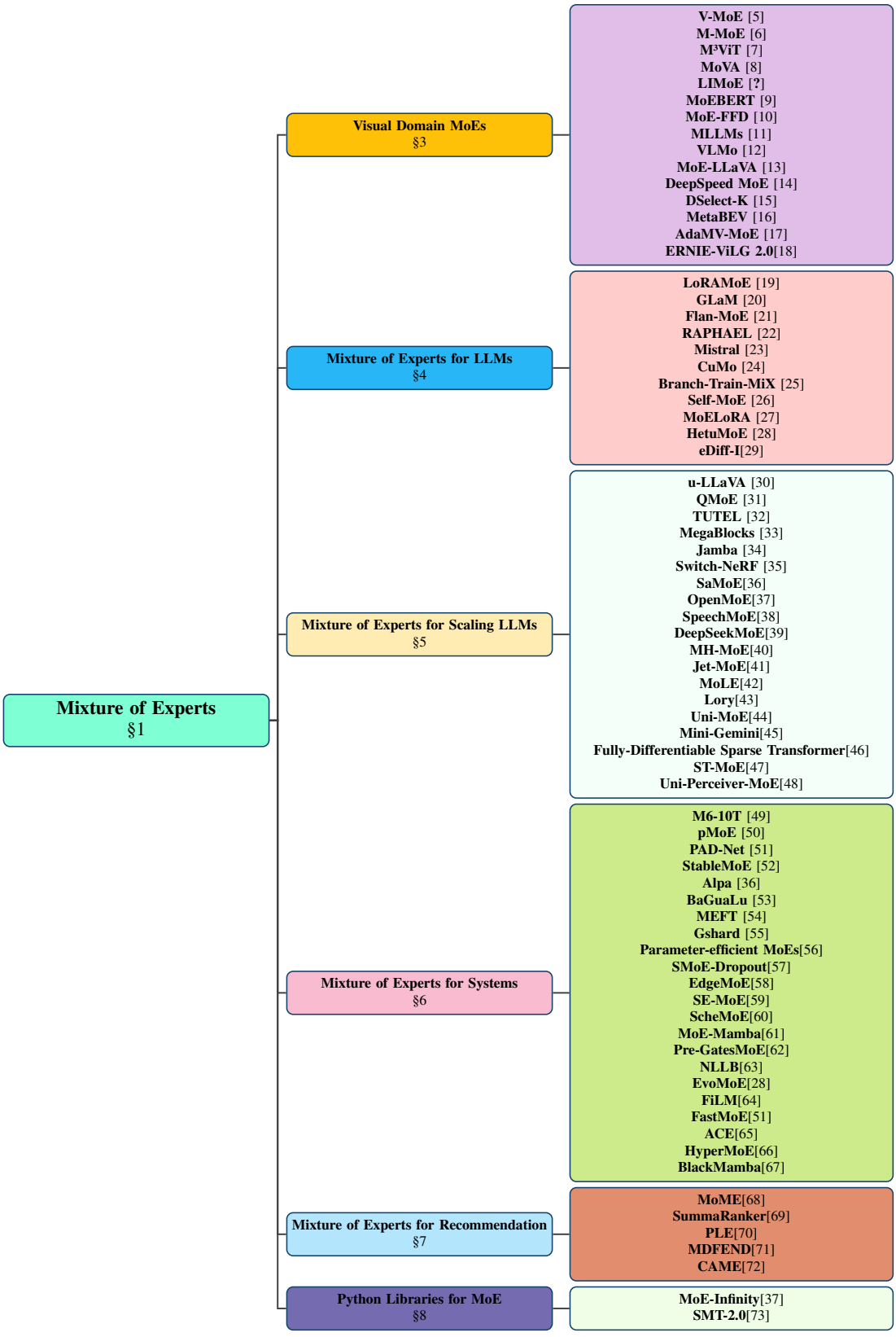

Fig. 3. Taxonomy of Mixture of Experts, encompassing Mixture of Experts for Vision,Mixture of Experts for LLMs, Mixture of experts on scaling the LLMs. Mixture of experts in recommendation system and various MoE Python Libraries.

2.3. MegaBlocks Handling of Load Balancing

MegaBlocks is a specific approach designed to handle load balancing more efficiently in MoEs. Here's how MegaBlocks addresses this issue:

1. **Block-wise Parallelism**
    - The model is divided into smaller, manageable blocks.
    - Each block can be processed in parallel, which helps distribute the computational load more evenly.
2. **Sparse Activation**
    - MegaBlocks uses structured sparsity to activate only a subset of the model for each input.
    - This reduces the computational cost and ensures that no single expert or block is overwhelmed with too much load.
3. **Efficient Load Distribution Algorithms**
    - Advanced algorithms are employed to monitor and adjust the load distribution in real-time.
    - These algorithms ensure that experts within each block are utilized evenly, preventing any single expert from becoming a bottleneck.
4. **Dynamic Capacity Adjustment**
    - MegaBlocks can dynamically adjust the capacity of each expert based on the current load.
    - This dynamic adjustment helps in redistributing the workload as needed, maintaining a balanced utilization of experts

In summary, load balancing in MoEs is crucial for efficient model performance. Incorporating load balancing terms in the loss function, using sophisticated gating mechanisms, and employing advanced routing strategies are some ways to address this issue. MegaBlocks, in particular, offers a robust solution by combining block-wise parallelism, structured sparsity, and efficient load distribution algorithms to ensure balanced expert utilization.

2.4. Expert Specialization

Recent research has uncovered some insights regarding how experts specialize within an MoE architecture.Shown below is a neat visualization from Towards Understanding the Mixture-of-Experts Layer in Deep Learning by Chen *et al*. [74], which shows how a 4-expert MoE model learns to solve a binary classification problem on a toy dataset that's segmented into 4 clusters. Initially, the experts (shown as different colors) are all over the place, but as training proceeds, different experts "specialize" in different clusters until there's almost a 1:1 correspondence. That specialization is entirely random, and only driven by the small initial random perturbations. Meanwhile, the gate is learning to (1) cluster the data and (2) map experts to clusters. One of the important take-away from this toy experiment is that non-linearity appears to be the key to the success of MoE. Experts with linear activation simply don't work as well as those with non-linear (cubic in this work) activation.

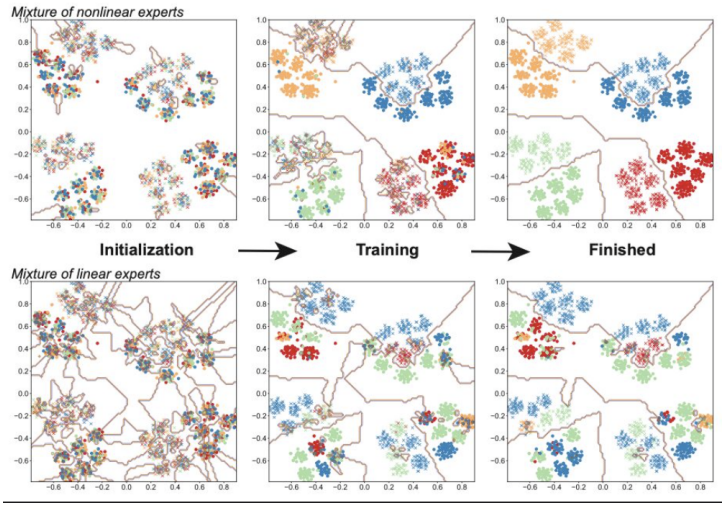

Fig. 4. Recent research has started to give us some insights. Here's a neat visualization from the paper "Towards Understanding the Mixture-of-Experts Layer in Deep Learning" by [74], which shows how a 4-expert MoE model learns to solve a binary classification problem on a toy dataset that's segmented into 4 clusters.

2.5. Token Dropping

In MoEs, expert capacity and capacity factor are crucial for ensuring balanced load distribution and efficient utilization of the model's experts. Expert capacity sets the maximum number of tokens or inputs each expert can process, while the capacity factor adjusts this capacity to provide flexibility in managing computational resources.

- **Causes of Token Dropping**
    1. **Imbalanced Expert Activation**:When the gating network disproportionately routes most tokens to a few experts, some tokens may end up not being assigned to any expert, effectively getting "dropped".
    2. **Capacity Constraints**: If the model enforces strict capacity limits on how many tokens each expert can handle, some tokens might be left out when these limits are reached.
    3. **Inefficient Gating Mechanism**:A gating mechanism that does not adequately distribute tokens across experts can cause some tokens to be neglected.
- **Mitigation Strategies for Token Dropping**
    1. **Enhanced Gating Mechanism**:Improve the gating network to ensure a more balanced distribution of tokens across experts.Use soft gating mechanisms that allow for more flexible expert selection, reducing the chances of token dropping.
    2. **Regularization Techniques**: Incorporate regularization terms in the loss function that penalize the model for uneven expert activation or for dropping tokens.Use load balancing regularization to ensure a more uniform token distribution.
    3. **Dynamic Capacity Allocation**:Allow for dynamic adjustment of the capacity of each expert based on the current load.Implement adaptive capacity constraints to

prevent experts from becoming overloaded while ensuring that all tokens receive attention.

4. **Token Routing Strategies**:Develop advanced token routing algorithms that ensure every token is assigned to at least one expert.Use probabilistic routing to distribute tokens more evenly among experts.

- **Specific Approaches to Mitigate Token Dropping**

1. **Auxiliary Loss Functions**:Introduce auxiliary loss functions that specifically penalize token dropping. For instance, a penalty can be added for tokens that do not get assigned to any expert.
2. **Token Coverage Mechanisms**: Implement token coverage mechanisms that ensure every token is attended to by at least one expert. This can involve ensuring that the sum of the gating probabilities for each token is above a certain threshold.
3. **MegaBlocks Approach**:MegaBlocks can help mitigate token dropping by utilizing block-wise parallelism and structured sparsity. Each block is responsible for a subset of the model's computation, and by structuring the activation sparsity, it ensures that all tokens are processed by some experts within the active blocks.
4. **Diversity Promoting Techniques**:Use techniques that promote diversity in expert selection. This can include using entropy maximization in the gating decisions to ensure a more diverse set of experts is activated for different tokens.
5. **Regular Monitoring and Adjustment**:Continuously monitor the distribution of tokens across experts and adjust the gating network or capacity constraints accordingly.Use feedback mechanisms to dynamically adjust the gating probabilities and ensure a balanced expert activation.

3. VISUAL DOMAIN-SPECIFIC IMPLEMENTATIONS OF MIXTURE OF EXPERTS

V-MoE: Almost all prevalent computer vision models networks are "dense," that is, every input is processed by every parameter.This paper by Riquelme *et al*. [5] from Google Brain introduces the Vision Mixture of Experts (V-MoE), a novel approach for scaling vision models. The V-MoE is a sparsely activated version of the Vision Transformer (ViT) that demonstrates scalability and competitiveness with larger dense networks in image recognition tasks.The paper proposes a sparse variant of the ViT that uses a MoE architecture. This approach routes each image patch to a subset of experts, making it possible to scale up to 15B parameters while matching the performance of state-of-the-art dense models.An innovative extension to the routing algorithm is presented, allowing prioritization of subsets of each input across the entire batch. This adaptive per-image compute leads to a trade-off between performance and computational efficiency during inference. The V-MoE shows impressive scalability, successfully trained up to 15B parameters, and demonstrates strong performance, including 90.35% accuracy on ImageNet.The paper explores the transfer learning abilities of V-MoE, showing its adaptability and effectiveness across different tasks and datasets, even with limited data.A detailed analysis of the V-MoE's routing decisions and the behavior of its experts is provided, offering insights into the model's internal workings and guiding future improvements.V-MoE models require less computational resources than dense counterparts, both in training and inference, thanks to their sparsely activated nature and the efficient use of the Batch Prioritized Routing algorithm.The paper represents a significant advancement in the field of computer vision, particularly in the development of scalable and efficient vision models.

MMoE: This paper by Jiaqi *et al*. [6] published in KDD 2018, introduces a novel approach to multi-task learning called Multi-gate Mixture-of-Experts (MMoE). The method aims to enhance the performance of multi-task learning models by better handling the relationships between different tasks.The MMoE model adapts the Mixture-of-Experts (MoE) framework to multi-task learning by sharing expert submodels across all tasks and using a gating network optimized for each task. This design allows the model to dynamically allocate shared and task-specific resources, efficiently handling tasks with varying degrees of relatedness.The paper presents experiments using synthetic data and real datasets, including a binary classification benchmark and a large-scale content recommendation system at Google. These experiments demonstrate MMoE's effectiveness in scenarios where tasks have low relatedness and its superiority over traditional shared-bottom multi-task models in terms of both performance and trainability.MMoE's architecture consists of multiple experts (feed-forward networks) and a gating network for each task, which determines the contribution of each expert to the task. This setup allows the model to learn nuanced relationships between tasks and allocate computation resources more effectively.On synthetic data, MMoE showed better performance, especially when task correlation is low, and demonstrated improved trainability with less variance in model performance across runs. On real-world datasets, including the UCI Census-income dataset and Google's content recommendation system, MMoE consistently outperformed baseline models in terms of accuracy and robustness.MMoE offers computational efficiency by using lightweight gating networks and shared expert networks, making it suitable for large-scale applications. The experiments on Google's recommendation system highlighted MMoE's ability to improve both engagement and satisfaction metrics in live experiments compared to single-task and shared-bottom models.

M³ViT: In the paper by Liang *et al*. [7] proposes Mixture-of-Experts Vision Transformer for Efficient Multi-task Learning with Model-Accelerator Co-design (M³ViT)framework presents a solution through a co-design approach between the model and its hardware accelerator. It leverages a MoE architecture embedded within a vision transformer. Imagine a team of specialists – that's the concept behind MoE. M³ViT has multiple sub-networks (experts) within the model, each specializing in handling specific visual aspects of the data. During training, a routing mechanism acts like a dispatcher, intelligently selecting only the most relevant expert(s) for each training sample. This sparse activation helps avoid training conflicts by disentangling the parameter spaces for different tasks, potentially leading to faster and more efficient training.

MoVA: In this paper by Zong *et al*. [8], proposes MoVA, a robust and innovative MLLM. MoVA employs a dynamic routing and fusion mechanism to leverage task-specific vision experts through a coarse-to-fine approach.During the coarse-grained phase, MoVA employs a context-aware expert routing strategy that dynamically selects the most suitable vision experts based on user instructions, input images, and the expertise of said vision experts. This approach harnesses the potent function understanding capabilities of the LLM equipped with expert-routing low-rank adaptation (LoRA), thereby enhancing overall model performance.Transitioning to the fine-grained phase, MoVA employs the mixture-of-vision-expert

adapter (MoV-Adapter), which meticulously extracts and fuses task-specific knowledge from a diverse array of experts. By adopting this coarse-to-fine paradigm, MoVA effectively utilizes expert representations based on multimodal context and model proficiency, thereby bolstering its generalization ability.

LIMoE: In this paper by Mustafa *et al*. [75] proposes a novel approach for multimodal learning that tackles this limitation. LIMoE leverages the efficiency of sparse MoE models by processing both images and text simultaneously. During training, a contrastive loss function helps the model learn similar representations for paired image-text data. The MoE architecture is particularly well-suited for this task, as different "expert" sub-networks can specialize in handling either images or text. The paper acknowledges challenges like training stability and balanced expert utilization, proposing an entropy-based regularization scheme to address them. LIMoE demonstrates impressive performance gains compared to traditional models with similar computational cost. It even achieves state-of-the-art zero-shot ImageNet accuracy when scaled appropriately. Interestingly, the study reveals that LIMoE can organically develop experts specializing in each modality, and the model exhibits varying treatment of images and text during processing. Overall, LIMoE offers a promising direction for efficient and effective multimodal learning.

MoEBERT:In this paper by Zuo *et al*. [9] proposes MoEBERT, a novel approach that leverages a Mixture-of-Experts (MoE) structure to achieve both high performance and fast inference speeds. MoEBERT starts by adapting the feed-forward neural networks within a pre-trained LLM into multiple experts. Here, the key concept is importance-based selection. The paper argues that not all neurons within the feed-forward network contribute equally to the model's performance. MoEBERT identifies the most important neurons and shares them across all experts, ensuring the preservation of the pre-trained model's representational power. The remaining neurons are then distributed evenly among the experts, promoting diversity and further enhancing model capabilities.During inference, MoEBERT employs a gating mechanism to activate only the most suitable expert for a specific task. This significantly improves efficiency compared to traditional methods that activate the entire LLM, even for simple tasks. Additionally, the paper proposes a layer-wise distillation method specifically designed for training MoEBERT. This method ensures that the knowledge from the pre-trained LLM is effectively transferred to the individual experts.The effectiveness of MoEBERT is then evaluated on natural language understanding and question answering tasks. The results demonstrate that MoEBERT surpasses existing task-specific distillation techniques. For instance, on the MNLI (mismatched) dataset, MoEBERT outperforms previous approaches by over 2%.In essence, this research presents MoEBERT as a promising solution for overcoming the trade-off between performance and efficiency in large language models. MoEBERT leverages a Mixture-of-Experts architecture with importance-based selection and a gating mechanism to achieve significant efficiency gains during inference while preserving the performance of the original LLM.

MoE-FFD: In this paper by Kong *et al*. [10] introduces a novel ViT-based framework designed to address the challenges of face forgery detection. The proposed MoE-FFD method integrates MoE modules with Low-Rank Adaptation(LoRA) [76] and Adapter layers to create a parameter-efficient model that effectively captures both global and local forgery clues. By using the Vision Transformer (ViT) as a frozen backbone and only updating lightweight layers, MoE-FFD significantly reduces computational and storage requirements. The MoE modules dynamically selects the most relevant experts, enhancing the model's capacity and detection performance. Rigorous experiments on multiple deepfake datasets demonstrate that MoE-FFD achieves state-of-the-art performance with a minimal parameter overhead, offering a robust and adaptable solution for real-world face forgery detection scenarios.

MLLMs: In this papaer by Wang *et al*. [11] introduces Visualized In-Context Text Processing (VisInContext), a novel method designed to address the challenge of processing long text contexts in multi-modality large language models (MLLMs) efficiently. By converting long in-context text into visual tokens, VisInContext significantly reduces GPU memory usage and computational costs, enabling an increase in the pre-training in-context text length from 256 to 2048 tokens with minimal additional floating point operations. This method, implemented within a Flamingo-based architecture, not only enhances in-context few-shot evaluation performance but also improves document understanding capabilities, showing promise in tasks like document question answering and sequential document retrieval.

VLMo:In this paper by Bao *et al*. [12] introducesstagewise pre-training approach that effectively leverages extensive image-only, text-only, and image-text paired datasets. Their experimental results showcase the remarkable performance of VLMo across diverse vision-language tasks, including VQA, NLVR2, and image-text retrieval, positioning it as a state-of-the-art solution. Additionally, they propose a versatile multimodal Transformer, dubbed MOME Transformer, tailored for vision-language tasks. This Transformer adeptly encodes various modalities by incorporating modality-specific information through dedicated experts and aligning the contents of different modalities via a shared self-attention module.

MoE-LLaVA: The paper by Lin *et al*. [13] introduces MoE-LLaVA, a novel training strategy for Large Vision-Language Models (LVLMs) called MoE-tuning. This approach constructs a sparse model with a large parameter count while maintaining constant computational costs, effectively addressing performance degradation in multi-modal learning and model sparsity. MoE-LLaVA activates only the top-k experts through routers during deployment, reducing hallucinations in model outputs. Impressively, it performs comparably to existing models with fewer parameters and outperforms others in object hallucination benchmarks. The architecture includes vision encoders, visual projection layers, word embedding layers, LLM blocks, and MoE blocks. MoE-tuning involves three stages: MLP training, training all parameters except the Vision Encoder, and initializing experts in MoE using FFNs and training only MoE layers. Evaluation on various datasets demonstrates MoE-LLaVA's efficiency and effectiveness, with performance matching or surpassing state-of-the-art models. The paper includes ablation studies and visualizations to illustrate the efficacy of MoE-tuning and MoE-LLaVA. Overall, it offers significant contributions to multi-modal learning systems, providing insights for future research in developing more efficient and effective models.

DeepSpeed MoE:In this paper by Rajbhandari *et al*. [14] introduces DeepSpeed MoE,that tackles the challenges of Mixture of Experts (MoE) models in both training and inference. This comprehensive solution introduces new MoE architectures and compression techniques, significantly reducing model size and improving efficiency. The system achieves remarkable performance gains, offering faster and more cost-effective inference compared to traditional dense models and existing MoE solutions. By enabling the deployment of high-quality, resource-efficient MoE models, DeepSpeed-MoE aims to shift the paradigm in large-scale AI modeling. This advancement opens up new possibilities for researchers and practitioners to explore and utilize sparse MoE models across various applications,

potentially transforming the landscape of large language models and AI systems.

DSelect-k:In this paper by Hazimeh *et al*. [15] proposes a novel approach to the MoE architecture, specifically designed to address challenges in multi-task learning scenarios. The key innovation is a differentiable top-k selection mechanism that allows for more efficient and effective expert selection in MoE models. It also allows for dynamic task-specific expert selection. DSelect-k provides a more efficient alternative to full MoE models, especially in scenarios with a large number of experts.

MetaBEV: In this paper by Ge *et al*. [16]introduces a novel Bird's Eye View (BEV) perception framework designed for both 3D object detection and BEV map segmentation. The framework incorporates M^2oE (Multimodal Mixture of Experts) structures to address task conflicts that arise when performing 3D detection and segmentation tasks using shared weights. A key component of MetaBEV is its robust fusion module, which features a new M^2oE-FN (Feed-Forward Network) layer. This innovative layer is specifically designed to mitigate gradient conflicts between detection and segmentation tasks, resulting in more balanced performance across both tasks. Notably, MetaBEV is the first framework to apply the MoE concept to 3D object detection and BEV map segmentation, offering a multi-modal, multi-task, and robust approach to autonomous driving perception challenges.

AdaMV-MoE: In this paper by Chen *et al*. [17]proposes an adaptive multi-task vision recognition framework designed to automatically adjust the network capacity for different tasks. This approach customizes the state-of-the-art Mixture-of-Experts (MTL MoE) models by optimizing task-specific model sizes through the adaptive activation or deactivation of experts. The framework automatically determines the number of activated experts for each task based on training dynamics, eliminating the need for laborious manual tuning of the optimal model size.

ERNIE-ViLG 2.0: In this paper by Feng *et al*. [18] proposes ERNIE-ViLG 2.0, a sophisticated Chinese text-to-image diffusion model, a significant advancement in image generation technology. The authors have developed this model to progressively enhance the quality of generated images through two key innovations: the incorporation of fine-grained textual and visual knowledge of crucial scene elements, and the utilization of specialized denoising experts at different stages of the denoising process. These improvements have yielded impressive results, with ERNIE-ViLG 2.0 setting a new state-of-the-art benchmark on the MS-COCO dataset, achieving a zero-shot FID-30k score of 6.75.

4. HARNESSING EXPERT NETWORKS FOR ADVANCED LANGUAGE UNDERSTANDING

LoRAMoE: This paper by Dou *et al*. [19] from Fudan University and Hikvision Inc, this paper introduces LoRAMoE, a new approach to alleviate the conflict between expanding supervised fine-tuning (SFT) data and retaining world knowledge in LLMs. The paper demonstrates that extensive SFT can disrupt world knowledge in LLMs, leading to knowledge forgetting.LoRAMoE is an innovative plugin version of Mixture of Experts (MoE), designed to preserve world knowledge by freezing the backbone model's parameters during training. It uses localized balancing constraints to coordinate expert groups, dividing them between task-specific learning and maintaining world knowledge.The architecture of LoRAMoE includes multiple parallel plugins as experts in each feed-forward layer of the LLM, connected by routers. These experts are divided into groups, where one focuses on downstream tasks and the other on aligning world knowledge with human instructions. This approach effectively reduces knowledge forgetting and improves downstream task performance.During the training process, only the experts and the router are optimized.Experiments conducted on various datasets demonstrate LoRAMoE's ability to manage experts based on data type and its effectiveness in preventing knowledge forgetting. The method shows improvement in downstream tasks, indicating its potential for multi-task learning. The visualization of expert utilization confirms that LoRAMoE effectively specializes experts for different types of tasks.The paper's innovative approach leverages the strengths of MoE and Low-Rank Adaptation (LoRA) for efficient training. It strategically addresses the issue of balancing expert utilization and preventing knowledge forgetting in LLMs, making it a notable contribution to the field of language model alignment.

GLaM: Scaling language models with more data, compute and parameters has driven significant progress in natural language processing. For example, thanks to scaling, GPT-3 was able to achieve strong results on in-context learning tasks. However, training these large dense models requires significant amounts of computing resources.This paper by Du *et al*. [20] in ICML 2022 proposes and develops a family of language models named GLaM (Generalist Language Model), which uses a sparsely activated mixture-of-experts architecture to scale the model capacity while also incurring substantially less training cost compared to dense variants. The largest GLaM has 1.2 trillion parameters, which is approximately 7x larger than GPT-3.It consumes only 1/3 of the energy used to train GPT-3 and requires half of the computation FLOPs for inference, while still achieving better overall zero-shot and one-shot performance across 29 NLP tasks.

Flan-MoE:The paper Mixture-of-Experts Meets Instruction Tuning by Shen *et al*. [21] presents significant advancements in the scalability and efficiency of LLMs through the novel integration of MoE architecture and instruction tuning, setting new standards in the field of natural language processing. Sparse MoE is a neural architecture that adds learnable parameters to LLMs without increasing inference costs. In contrast, instruction tuning trains LLMs to follow instructions more effectively.The authors advocate for the combination of these two approaches, demonstrating that MoE models benefit significantly more from instruction tuning compared to their dense model counterparts.The paper presents three experimental setups: direct finetuning on individual downstream tasks without instruction tuning; instruction tuning followed by few-shot or zero-shot generalization on downstream tasks; and instruction tuning supplemented by further finetuning on individual tasks.The findings indicate that MoE models generally underperform compared to dense models of the same computational capacity in the absence of instruction tuning. However, this changes with the introduction of instruction tuning, where MoE models outperform dense models.The paper introduces the FLAN-MOE32B model, which outperforms FLAN-PALM62B on four benchmark tasks while using only a third of the FLOPs. This highlights the efficiency and effectiveness of the FLAN-MOE approach.The authors conduct a comprehensive series of experiments to compare the performance of various MoE models subjected to instruction tuning. These experiments include evaluations in natural language understanding, reasoning, and question-answering tasks. The study also explores the impact of different routing strategies and the number of experts on the performance of FLAN-MOE models, showing that performance scales with the number of tasks rather than the number of experts.The following image from the paper shows the effect of instruction tuning on MOE models versus dense counterparts for base-size models. They perform single-task fine-tuning for each model on held-out benchmarks. Compared to dense

models, MoE models benefit more from instruction-tuning, and are more sensitive to the number of instruction-tuning tasks. Overall, the performance of MoE models scales better with respect to the number of tasks, than the number of experts.The paper discusses the challenge of adapting MoE models to multilingual benchmarks and highlights the importance of incorporating diverse linguistic data during training to ensure effective language coverage.

RAPHAEL: This paper by Xue *et al*. [22] from the University of Hong Kong and SenseTime Research, the authors introduce RAPHAEL, a novel text-to-image diffusion model that generates highly artistic images closely aligned with textual prompts. RAPHAEL uniquely combines tens of MoEs layers, including space-MoE and time-MoE layers, allowing billions of diffusion paths. Each path intuitively functions as a "painter" for depicting specific textual concepts onto designated image regions at certain diffusion timesteps. This mechanism substantially enhances the precision in aligning text and image content.The authors report that RAPHAEL outperforms recent models like Stable Diffusion, ERNIE-ViLG 2.0, DeepFloyd, and DALL-E 2 in terms of image quality and aesthetic appeal. This is evidenced by superior performance in diverse styles (e.g., Japanese comics, realism, cyberpunk) and a state-of-the-art zero-shot FID score of 6.61 on the COCO dataset.An edge-supervised learning module is introduced to further refine image quality, focusing on maintaining intricate boundary details in various styles. RAPHAEL is implemented using a U-Net architecture with 16 transformer blocks, each containing a self-attention layer, a cross-attention layer, space-MoE, and time-MoE layers. The model, with three billion parameters, was trained on 1,000 A100 GPUs for two months. Framework of RAPHAEL. (a) Each block contains four primary components including a selfattention layer, a cross-attention layer, a space-MoE layer, and a time-MoE layer. The space-MoE is responsible for depicting different text concepts in specific image regions, while the time-MoE handles different diffusion timesteps. Each block uses edge-supervised cross-attention learning to further improve image quality. (b) shows details of space-MoE. For example, given a prompt "a furry bear under sky", each text token and its corresponding image region (given by a binary mask) are directed through distinct space experts, i.e., each expert learns particular visual features at a region. By stacking several space-MoEs, we can easily learn to depict thousands of text concepts.The authors conducted extensive experiments, including a user study using the ViLG-300 benchmark, demonstrating RAPHAEL's robustness and superiority in generating images that closely conform to the textual prompts. The study also showcases RAPHAEL's flexibility in generating images of diverse styles and high resolutions up to 4096 × 6144 when combined with a tailor-made SR-GAN model.RAPHAEL's potential applications extend to various domains, with implications for both academic research and industry. The model's limitations include the potential misuse for creating misleading or false information, a challenge common to powerful text-to-image generators

Mistral: Mistral 8x7B (56B params) by Jiang *et al*. [23] from Mistral follows a MoE architecture, consisting of 8x 7B experts. With 8 experts and a router network that selects two of them at every layer for the inference of each token, it looks directly inspired from rumors about GPT-4's architecture. From GPT-4 leaks, we can speculate that GPT-4 is a MoE model with 8 experts, each with 111 B parameters of their own and 55B shared attention parameters (166B parameters per model). For the inference of each token, also only 2 experts are used. Since the model size (87GB) is smaller than 8x Mistral 7B (8 × 15GB=120GB), we could assume that the new model uses the same architecture as Mistral 7B but the attention parameters are shared, reducing the naïve 8x7B model size estimation.The conclusion is that (probably) Mistral 8x7B uses a very similar architecture to that of GPT-4, but scaled down to 8 total experts instead of 16 (2× reduction), 7B parameters per expert instead of 166B (24× reduction), 42B total parameters (estimated) instead of 1.8T (42× reduction), free to use under Apache 2.0 license, Outperforms Llama 2 70B with 6x faster inference. Matches or outperforms GPT-3.5, Multilingual: vastly outperforms `LLaMA 2` [77] 70B on French, Italian, German and Spanish, Same 32K context as the original GPT-4. Each layer in a 8x MoE model has its FFN split into 8 chunks and a router picks 2 of them, while the attention weights are always used in full for each token. This means that if the new mistral model uses 5B parameters for the attention, you will use $5 + (42 - 5)/4 = 14.25B$ params per forward pass. Mixtral is basically 8 models in a trenchcoat: the feedforward layers of the decoder blocks are divided into 8 experts, and for each token, a router will decide which 2 experts to allocate the processing to. The advantage of this architecture is that even though you have $78B = 47B$ parameters in total (considering shared parameters which are not unique to each expery), the model is much cheaper and fast to run since only 28 experts are activated for each prediction.

CuMo: In the paper by Li *et al*. [24] proposes a method CuMo, incorporates Co-upcycled Top-K sparsely-gated MoE blocks into both the vision encoder and the MLP connector, thereby enhancing multimodal LLMs with minimal additional activated parameters during inference. CuMo first pre-trains the MLP blocks and then initializes each expert in the MoE block from the pre-trained MLP block during the visual instruction tuning stage, using auxiliary losses to ensure balanced expert loading. This approach allows CuMo to outperform state-of-the-art multimodal LLMs on various VQA and visual-instruction-following benchmarks across different model size groups, while training exclusively on open-source datasets.

Branch-Train-MiX: The paper by Sukhbaatar *et al*. [25] introduces Mixing Expert LLMs into a Mixture-of-Experts LLM that presents a method called Branch-Train-MiX (BTX) designed to efficiently train LLMs across multiple specialized domains. BTX begins with a seed model, which is branched into multiple copies that are trained independently on different datasets to become domain-specific experts. These expert models are then combined into a single model using Mixture-of-Experts (MoE) layers, with the remaining parameters averaged and finetuned to optimize performance. This approach leverages the benefits of parallel training to reduce communication costs and increase throughput, while the MoE framework ensures efficient parameter utilization and performance across various tasks. Experimental results using the Llama-2 7B model demonstrate that BTX achieves superior accuracy and efficiency compared to other methods, particularly in specialized domains such as mathematics and code, without suffering from catastrophic forgetting.

Self-MoE : The paper by Kanf *et al*. [26] introduces Self-MoE, a method to transform monolithic LLMs into a modular system called MiXSE (MiXture of Self-specialized Experts). This approach leverages self-specialization using synthetic data and self-optimized routing, enabling dynamic and specific task handling without extensive human-labeled data or additional parameters. The experiments reveal significant improvements in performance across various benchmarks, such as knowledge, reasoning, math, and coding, without compromising non-targeted domains. Self-MoE's modular design enhances flexibility, interpretability, and adaptability, outperforming strong baselines like instance and weight merging. The method highlights the critical role of the routing mechanism and semantic experts and is universally applicable across different model families

and sizes. The paper concludes that Self-MoE effectively enhances LLMs' efficiency, scalability, and adaptability, marking a significant advancement in modular and self-improving LLM specialization techniques, with plans to release the code for broader application and validation.

MOELoRA: In this paper by Liu *et al*. [27] proposes a novel multi-task Parameter-Efficient Fine-Tuning (PEFT) framework that leverages the strengths of both Mixture-of-Experts (MoE) and Low-Rank Adaptation (LoRA). It introduces a task-motivated gate function to facilitate tuning distinct parameter sets for each task. The authors claim this work is the first to explore multi-task PEFT techniques for large language model-driven medical applications, addressing the unique challenges of parameter efficiency and task-specific tuning in this context.

HetuMoE: In this paper by Nie *et al*. [28] proposes a novel system designed to handle the challenges of training large-scale MoE models. HetuMoE introduces efficient mechanisms to manage and distribute the computational load across multiple processing units, ensuring scalability to trillion-scale parameters. The system incorporates advanced techniques to optimize both memory usage and computational efficiency, making it suitable for extensive machine learning tasks. This framework enables more effective and efficient training of large-scale models, paving the way for advancements in deep learning and artificial intelligence

eDiff-I: In this paper by Balaji *et al*. [29] proposes an ensemble-of-expert-denoisers design to enhance the quality of text-to-image generation while maintaining the same inference computation cost. The expert denoisers are trained using a finetuning scheme that reduces training costs. Additionally, the paper suggests using an ensemble of encoders, including the T5 text encoder, the CLIP text encoder, and the CLIP image encoder, to provide input information to the diffusion model. The text encoders favor different image formations, and the CLIP image encoder allows for style transfer using a reference photo. The paper also introduces a training-free extension that enables paint-with-words capability through a cross-attention modulation scheme, giving users additional spatial control over the text-to-image output.

5. LLM EXPANSION VIA SPECIALIZED EXPERT NETWORKS

u-LLaVA: In this paper by Jinjin [30] have introduced have introduced a novel framework called u-LLaVA. This framework aims to refine the MLLMs' perception by integrating information from three levels of detail: pixels, regions, and the entire image. Imagine u-LLaVA as a detective – it can analyze individual clues (pixels), understand the relationships between them (regions), and grasp the overall scene (global features) to form a more complete picture.The foundation for strong visual understanding is laid by training u-LLaVA on both image and video data. Images provide a rich set of features, while videos introduce the element of time and a wider range of visual contexts. By leveraging both data types, u-LLaVA builds a robust understanding of diverse visual scenarios. Overall, u-LLaVA represents a significant leap forward in multi-modal learning. By integrating multi-level visual information, efficient modality alignment, and task-specific instruction tuning, it pushes MLLMs beyond broad comprehension and equips them for robust perception of visual details. This opens doors for various applications that require a nuanced understanding of visual content, such as generating detailed image captions, answering visual questions that involve specific objects or locations, and more.

QMoE: In this paper by Frantar *et al*. [31] introduces QMoE, a framework aimed at addressing the memory challenges associated with deploying LLMs using MoE architectures.The primary issue QMoE tackles is the substantial memory requirement of large models, exemplified by the 1.6 trillion-parameter SwitchTransformer-c2048 model, typically demanding 3.2TB of memory. QMoE effectively compresses such models to less than 1 bit per parameter, allowing their execution on commodity hardware with minimal runtime overhead.Employing a scalable algorithm and a custom compression format paired with GPU decoding kernels, QMoE compresses the SwitchTransformer-c2048 model to less than 160GB (0.8 bits per parameter) with minor accuracy loss in under a day on a single GPU.The implementation encompasses a highly scalable compression algorithm and a bespoke compression format, facilitating efficient end-to-end compressed inference. This framework enables the operation of trillion-parameter models on affordable hardware, such as servers equipped with NVIDIA GPUs, with less than 5% runtime overhead compared to ideal uncompressed execution.The paper discusses challenges in compressing MoE models, including conceptual issues with existing post-training compression methods and practical scaling challenges. QMoE overcomes these by introducing a custom compression format and highly efficient decoding algorithms optimized for GPU accelerators.Technical contributions include a novel approach to handling massive activation sets and a unique system design for optimized activation offloading, expert grouping, and robustness modifications, ensuring efficient application of data-dependent compression to massive MoEs.Significantly reducing the size of large models, QMoE compressed models achieve over 20x compression rates compared to 16-bit precision models. This reduction is accompanied by minor increases in loss on pretraining validation and zero-shot data.The paper also discusses system design and optimizations addressing memory costs, GPU utilization, and reliability requirements, including techniques like optimized activation offloading, list buffer data structures, lazy weight fetching, and expert grouping.Experiments demonstrate that QMoE effectively compresses MoE models while maintaining performance, tested on various datasets such as Arxiv, GitHub, StackExchange, and Wikipedia, showcasing good performance preservation even for highly compressed models.Providing detailed insights into encoding and decoding processes and kernel implementation for the GPU, the paper highlights challenges and solutions for achieving sub-1-bit per parameter compression.The QMoE framework represents a significant advancement in the practical deployment of massive-scale MoE models, addressing key limitations of MoE architectures and facilitating further research and understanding of such models.The paper's findings are crucial as they enable the deployment and research of trillion-parameter models on more accessible hardware, potentially democratizing access to high-performance LLMs and fostering further innovation in the field.

TUTEL: This paper by Hwang *et al*. [32] from MSR, published in MLSys, introduces TUTEL, a scalable and adaptive system designed to optimize the performance of sparsely-gated mixture-of-experts (MoE) models. The motivation behind TUTEL stems from the recognition that the dynamic nature of MoE models, which route input tokens to different experts based on a gating function, poses significant challenges for efficient computation due to the static execution strategies of existing systems. These strategies fail to adapt to the varying workload of experts, leading to inefficient use of computational resources.TUTEL addresses these challenges by introducing a dynamic execution framework that supports adaptive parallelism and pipelining. The system's design allows for the distribution of MoE model parameters and input data in an identical

layout, enabling seamless switching between parallelism strategies without the need for costly data or tensor migration. This capability facilitates real-time optimization of parallelism and pipelining during runtime, significantly enhancing computational efficiency.The authors implement various MoE acceleration techniques within TUTEL, including a Flexible All-to-All communication strategy, two-dimensional hierarchical (2DH) All-to-All, and fast encode/decode mechanisms. These innovations collectively enable TUTEL to deliver substantial speedups in the execution of single MoE layers over multiple GPUs, demonstrating improvements of 4.96x and 5.75x over the state-of-the-art on 16 and 2048 NVIDIA A100 GPUs, respectively.The evaluation of TUTEL showcases its effectiveness in running MoE-based models, specifically a real-world model named SwinV2-MoE, which is built upon the Swin Transformer V2 architecture. TUTEL significantly accelerates the training and inference of SwinV2-MoE, achieving speedups of up to 1.55x and 2.11x, respectively, over Fairseq, a previous framework. Moreover, the SwinV2-MoE model achieves superior accuracy in both pre-training and downstream computer vision tasks compared to its dense counterpart, highlighting TUTEL's practical utility in enabling efficient training and deployment of large-scale MoE models for real-world applications.

MegaBlocks:This paper by Gale *et al*. [33],introduces Dropless MoE, a novel system for efficient MoE training on GPUs. The system, named MegaBlocks, addresses the limitations of current frameworks that restrict dynamic routing in MoE layers, often leading to a tradeoff between model quality and hardware efficiency due to the necessity of dropping tokens or wasting computation on excessive padding. Token dropping leads to information loss, as it involves selectively ignoring part of the input data, while padding adds redundant data to make the varying input sizes uniform, which increases computational load without contributing to model learning. This challenge arises from the difficulty in efficiently handling the dynamic routing and load-imbalanced computation characteristic of MoE architectures, especially in the context of deep learning hardware and software constraints. MegaBlocks innovatively reformulates MoE computations as block-sparse operations, developing new GPU kernels specifically for this purpose. These kernels efficiently manage dynamic, load-imbalanced computations inherent in MoEs without resorting to token dropping. This results in up to 40% faster end-to-end training compared to MoEs trained with the Tutel library, and 2.4 times speedup over DNNs trained with Megatron-LM.The system's core contributions include high-performance GPU kernels for block-sparse matrix multiplication, leveraging blocked-CSR-COO encoding and transpose indices. This setup enables efficient handling of sparse inputs and outputs in both transposed and non-transposed forms.Built upon the Megatron-LM library for Transformer model training, MegaBlocks supports distributed MoE training with data and expert model parallelism. Its unique ability to avoid token dropping through block-sparse computation provides a fresh approach to MoE algorithms as a form of dynamic structured activation sparsity.The system also reduces the computational overhead and memory requirements associated with MoE layers, leading to more efficient utilization of hardware resources. Furthermore, the approach decreases the number of hyperparameters that need to be re-tuned for each model and task, simplifying the process of training large MoE models.The paper provides detailed insights into the design and performance of the block-sparse kernels, including analyses of throughput relative to cuBLAS batched matrix multiplication and discussions on efficient routing and permutation for MoEs. The results show that MegaBlocks' kernels perform comparably to cuBLAS, achieving an average of 98.6% of cuBLAS's throughput with minimal variations across different configurations.

Jamba: This paper by Lieber *et al*. [34] from AI21labs presents Jamba, an innovative architecture blending Transformer and Mamba layers with mixture-of-experts (MoE) modules, creating a synergistic model that excels in both performance and efficiency. This large language model is distinguished by its ability to process up to 256K tokens context length, designed to optimize computational resources by fitting within a single 80GB GPU using 8bit precision, showcasing 12B active and 52B total parameters. The architecture features Jamba blocks, each composed of a mix of Mamba and Transformer layers, interspersed with MoE layers. Jamba employs a configuration of four Jamba blocks encompassing a total of 4 Transformer and 28 Mamba layers, with an 8-layer structure per block and a strategic 1:7 Attention-to-Mamba ratio. MoE layers, placed every other layer, comprise 16 experts, utilizing the top 2 experts per token for dynamic adaptability.Implementation specifics include the use of grouped-query attention (GQA) and SwiGLU activation function within Transformer blocks, aiming for enhanced model stability and performance. A notable innovation is the addition of RMSNorm to Mamba layers for large-scale stability, effectively preventing loss spikes during training.Jamba's design eschews explicit positional information mechanisms like RoPE, relying instead on Mamba layers' implicit position encoding capabilities. This choice reflects insights from the model's development, suggesting that Mamba layers alone may sufficiently capture positional dependencies. MoE integration is demonstrated to significantly improve the hybrid Attention-Mamba model, underscoring MoE's contribution to enhancing the model's capacity and efficiency. This advancement is validated through extensive experimentation, although specific mechanisms behind MoE's effectiveness remain an area for further exploration.It is trained on NVIDIA H100 GPUs, utilizing Full-Model Sharded Data Parallelism (FSDP) along with Tensor, Sequence, and Expert Parallelism for optimal efficiency. The model leverages a comprehensive text dataset aggregated from web sources, books, and code, updated until March 2024, though detailed dataset size or the number of training tokens were not specified.Jamba achieves comparable or superior results against leading models such as Mixtral 8x7B and Llama-2 70B across a variety of benchmarks, especially in long-context evaluations. It is also noted for its remarkable throughput improvement, particularly for long contexts, compared to similar-sized attention-only models.In a move to foster further research and optimization within the community, Jamba is released under the Apache 2.0 license on HuggingFace. This initiative is supported by the release of model checkpoints from smaller-scale training runs, inviting wider exploration of the model's novel architecture and potential applications.Jamba exemplifies the potential of combining Transformer and Mamba architectures with MoE, setting new standards in language modeling for long-context processing while addressing computational and memory efficiency. Its development reflects significant technical advancements, promising to drive future research and applications in the field of natural language processing.

Switch-NeRF: This paper by Zhenxing *et al*. [35] proposes a MoE-based Switch-NeRF model which is carefully implemented and optimized to achieve both high-fidelity scene reconstruction and efficient computation. SWITCH-NERF tackles the challenge of large-scale scene modeling in Neural Radiance Fields (NeRFs) by proposing a learning-based scene decomposition with a MoE architecture. This approach decomposes complex scenes into smaller parts, assigning a dedicated sub-network (expert) to handle each. A gating network dynamically routes 3D points to the most suitable expert, improving efficiency and potentially enhancing rendering quality. While promising, the paper would benefit from discussions

on evaluation metrics, generalizability to diverse scenes, and exploration of MoE design choices for optimal configurations. Overall, SWITCH-NERF presents a significant step towards efficient and high-fidelity rendering of large-scale scenes using NeRFs.

SaMoE: This paper by Gao *et al*. [78] addresses a challenge in scaling LLMs efficiently. While MoE offers a way to create powerful LLMs with a vast number of parameters without significant computational burdens, there's a trade-off. Existing MoEs can be parameter-inefficient – simply adding more experts (sub-models) doesn't always guarantee better performance.The paper proposes a solution called SaMoE (Sparse Mixture-of-Experts) to address this parameter inefficiency. Their analysis reveals that as the number of experts in an MoE layer increases, the flow of information used for training (gradients) weakens. This weakness hinders the proper training of individual experts, limiting overall performance gains.SaMoE tackles this by introducing a new MoE architecture design. Instead of relying solely on a single "expert" for each input, SaMoE learns a soft combination of a global set of expert layers. Imagine a team of specialists – SaMoE allows the model to consult and combine insights from multiple experts, rather than relying on just one. This approach allows SaMoE to achieve significant parameter savings compared to standard MoE training.The paper showcases the effectiveness of SaMoE through extensive experiments on large, autoregressive MoE language models similar to GPT-3 (having billions of parameters). The results demonstrate that SaMoE can significantly improve parameter efficiency by reducing the total number of parameters by up to 5.2 times. Interestingly, SaMoE even achieves superior performance in pre-training and handling unseen tasks (zero-shot generalization) compared to the baseline MoE training approach.In essence, SaMoE offers a promising solution for building powerful LLMs with MoE architecture while ensuring efficient use of parameters. This allows for creating highly capable models without incurring the usual computational costs associated with massive parameter sizes.

OpenMoE: This paper by Xue *et al*. [37]introduces a significant contribution to the field. The core of this contribution is OpenMoE, a collection of open-source MoE LLMs. These models come in a range of sizes, from a relatively manageable 650 million parameters to a staggering 34 billion, all trained on massive datasets exceeding 1 trillion tokens. By making both the code and training data publicly available, the authors effectively democratize MoE technology, inviting researchers to explore and build upon their work.Beyond just accessibility, the paper explores the cost-effectiveness of MoE LLMs. The study suggests that MoE-based models offer a significant advantage over traditional, dense LLMs. While dense models can be powerful, they often require immense computational resources. MoE LLMs, on the other hand, seem to achieve a more favorable balance between effectiveness and cost. This makes them a promising direction for future LLM development, with the potential to lead to more efficient and powerful language models.However, the paper doesn't stop at simply introducing OpenMoE. The authors take a critical look under the hood of MoE models, specifically focusing on how they decide which "expert" sub-model handles a particular piece of text data (token).

SpeechMoE: In this paper by You *et al*. [38] proposes a novel approach called SpeechMoE, which combines MoE with Transformers. MoE offers an enticing benefit for large language models: it allows them to grow in capacity without a corresponding surge in computational cost. Additionally, MoE-based models can dynamically adapt to the varying complexities found in real-world speech data.SpeechMoE is specifically designed for this task and leverages two key improvements. First, a sparsity L1 loss function ensures efficient routing by controlling how often the gating mechanism activates different experts. Second, a mean importance loss promotes diversity in the gating mechanism's output, guaranteeing a wider range of experts are consulted for each speech input.Beyond these improvements, SpeechMoE boasts a new router architecture that utilizes information from two sources: a shared embedding network and the hierarchical representations generated by different MoE layers. Evaluations show that SpeechMoE significantly outperforms traditional static networks in terms of character error rate (CER) while maintaining similar computational requirements. This translates to CER improvements ranging from 7.0% to 23.0% on various benchmark datasets. Overall, SpeechMoE presents a promising direction for speech recognition by combining MoE's efficiency and adaptability with a well-designed architecture, leading to substantial gains in accuracy.

DeepSeekMoE: In this paper by Dai *et al*. [39] introduce DeepSeekMoE 16B, a MoE language model comprising 16.4B parameters. This model features an innovative MoE architecture, incorporating two primary strategies: fine-grained expert segmentation and shared expert isolation. Trained from scratch on 2T English and Chinese tokens, DeepSeekMoE 16B demonstrates performance comparable to DeekSeek 7B and LLaMA2 7B, while requiring only around 40% of the computational resources. For the benefit of the research community, they release the model checkpoints for DeepSeekMoE 16B Base and DeepSeekMoE 16B Chat to the public, enabling deployment on a single GPU with 40GB of memory without the need for quantization.DeepSeekMoE's architecture, particularly the use of shared experts, could lead to significant efficiency gains in various language processing tasks. By focusing on common knowledge, shared experts reduce redundancy and potentially allow other experts to specialize in more nuanced areas. This translates to faster processing times and potentially lower computational costs

MH-MoE: In this paper by Wu *et al*. [40] proposes Multi-Head Mixture-of-Experts (MH-MoE) model which utilizes a multi-head approach to divide each token into multiple sub-tokens. These sub-tokens are distributed among various experts, processed concurrently, and then seamlessly recombined into their original token form. This multi-head mechanism allows the model to collectively attend to information from different representation spaces within various experts, significantly enhancing expert activation. This process deepens context understanding and reduces overfitting. The MH-MoE model is simple to implement and operates independently of other SMoE optimization methods, making it easy to integrate with other SMoE models for improved performance.

JetMoE: In this paper by Shen *et al*. [41] proposes a high cost of training powerful Large Language Models (LLMs) by introducing JetMoE-8B, a model achieving impressive results for under $0.1 million. This demonstrates significant cost-effectiveness compared to traditional methods. JetMoE-8B utilizes a special architecture that activates only a portion of its parameters for each input, reducing computation by 70% compared to similar models. Additionally, JetMoE-8B surpasses existing models in performance, even outperforming a larger and presumably more expensive model in a chat-focused task. By openly sharing training data, code, and parameter details, the researchers promote JetMoE-8B as a foundation for future, cost-effective LLM development in academia.

MoLE: In this paper by Wu *et al*. [42] proposes a new multi-lingual speech recognition network, tackles a gap in current research. While existing methods focus on improving recognition accuracy across languages, MoLE goes a step further by also identifying the language being spoken. It achieves this by employing a lightweight language tokenizer that activates a specific language expert network

for the identified language. This expert is then combined with a general-purpose expert, weighted by the tokenizer's confidence in its identification. This "language-conditioned embedding" proves particularly effective in recognizing speech from languages with limited training data, making MoLE a valuable contribution for multi-lingual speech recognition, especially for low-resource languages.

Lory: In this paper by Zhong *et al*. [43], author proposes a novel MoE architecture tailored for autoregressive language model pretraining. This architecture is the first of its kind to be fully differentiable, enabling end-to-end gradient backpropagation without the need for load balancing objectives or intricate assignment algorithms. To enhance efficiency while maintaining autoregressiveness, the authors propose causal segment routing. Additionally, they introduce similarity-based data batching to promote expert specialization. Comparative evaluations reveal that Lory models featuring up to 32 experts consistently outperform dense models across various metrics, including perplexity and downstream tasks, achieving performance gains ranging from 1.5% to 13.9%. Notably, Lory demonstrates domain-level expert specialization, a departure from previous MoE language models that exhibit token-level routing and display more superficial, localized patterns. The findings underscore the promise of differentiable MoE architectures in the context of pretraining and call for further exploration in this area.

Uni-MoE: In this paper by Li *et al*. [44] proposes a innovative Uni-MoE model is a unified Multimodal Large Language Model (MLLM) based on the MoE architecture, designed to accommodate various modalities. It integrates modality-specific encoders into a cohesive multimodal representation and utilizes a sparse MoE structure within the LLMs to facilitate efficient training and inference through both modality-level data parallelism and expert-level model parallelism. To boost collaboration among experts and generalization, it adopts a progressive training strategy that includes aligning cross-modality data using different connectors, training modality-specific experts with cross-modality instruction data, and refining the Uni-MoE framework with Low-Rank Adaptation (LoRA) on mixed multimodal instruction data. Evaluation results on extensive multimodal datasets demonstrate that Uni-MoE effectively reduces performance bias in managing mixed multimodal datasets, while enhancing expert collaboration and generalization capabilities.

Mini-Gemini: In this paper by Li *et al*. [45] author proposes a novel framework for Vision Language Models (VLMs) that enhances high-resolution image processing, data quality, and application versatility. Mini-Gemini employs a dual-encoder system for efficient high-resolution processing, combining low-resolution embeddings with high-resolution detail refinement through ConvNet. The framework integrates high-quality data from diverse sources, including high-quality responses and task-oriented instructions, to improve model performance. Additionally, Mini-Gemini's any-to-any paradigm supports versatile applications, such as simultaneous image and text generation, leveraging advanced generative models. The framework sets new benchmarks in VLM capabilities, outperforming models like Gemini Pro and GPT-4V, particularly in complex datasets, and demonstrating strong instruction-following and reasoning abilities. Future work will focus on improving training data quality and exploring advanced visual reasoning and generation methods.

Fully-Differential Sparse Transformer:Sparse MoE architectures scale model capacity without large increases in training or inference costs. MoE allows us to dramatically scale model sizes without significantly increasing inference latency. In short, each "expert" can separately attend to a different subset of tasks via different data subsets before they are combined via an input routing mechanism. Thus, the model can learn a wide variety of tasks, but still specialize when appropriate. Despite their success, MoEs suffer from a number of issues: training instability, token dropping, inability to scale the number of experts, or ineffective finetuning. This paper by Puigcerver *et al*. [46] from Google DeepMind proposes Soft MoE, a fully-differentiable sparse Transformer that addresses these challenges, while maintaining the benefits of MoEs.Extra-large models like Google's PaLM (540B parameters) or OpenAI's GPT-4 use Sparse MoE under the hood, which suffers from training instabilities, because it's not fully differentiable. Soft-MoE replaces the non-differentiable expert routing with a differentiable layer. The end-to-end model is fully differentiable again, can be trained with ordinary SGD-like optimizers, and the training instabilities go away.Soft MoE performs an implicit soft assignment by passing different weighted combinations of all input tokens to each expert. As in other MoE works, experts in Soft MoE only process a subset of the (combined) tokens, enabling larger model capacity at lower inference cost.

ST-MoE: In this paper by Zoph *et al*. [47] introduces framework for creating sparse expert models that are both stable and transferable across tasks. The authors address instability issues in sparse models by proposing architectural modifications and training techniques that enhance robustness. They also focus on improving the transferability of these models to new domains and tasks without significant performance degradation. The results demonstrate that the proposed methods achieve better stability and transferability compared to existing approaches.

Uni-Perceiver-MoE: In this paper by Zhu *et al*. [48] intorduces Conditional Mixture-of-Experts (Conditional MoEs) for generalist models, proposing routing strategies under various conditions for both training and inference phases. Conditional MoEs are used to tackle the task-interference issue in generalist models by incorporating information about the current task and modalities. This approach effectively reduces interference while maintaining low computational and memory costs, as well as preserving generalization capabilities.

6. MOE: ENHANCING SYSTEM PERFORMANCE AND EFFICIENCY

M6-10T: In this paper by Lin *et al*. [49] introduce a training strategy called "Pseudo-to-Real" designed for large models that require significant memory. This strategy is suitable for models with sequential layers. They demonstrate the pretraining of a groundbreaking 10-trillion-parameter model, which is ten times larger than the current state-of-the-art, using only 512 GPUs over a span of 10 days. Alongside the Pseudo-to-Real method, they present a technique known as Granular CPU Offloading to effectively manage CPU memory and maintain high GPU utilization during the training of large models. This approach allows for efficient training of massive models on limited resources, significantly reducing the carbon footprint and promoting more environmentally friendly AI practices.

pMoE: In this paper by Chowdhury *et al*. [50] proposes a patch-level routing method in Mixture-of-Experts (pMoE) that divides each input into tokens and sends selected patches to experts through prioritized routing. This method has demonstrated considerable empirical success by reducing training and inference costs while maintaining test accuracy. Despite these empirical successes, the theoretical foundations of pMoE and general MoE models have been less clear. Focusing on a supervised classification task using a mixture of two-layer convolutional neural networks (CNNs), the authors demonstrate that pMoE can provably reduce the number of

training samples required to achieve desirable generalization (known as sample complexity) by a polynomial factor. Additionally, pMoE outperforms its single-expert counterpart of equal or even greater capacity. This advantage is attributed to the discriminative routing capability, which allows pMoE routers to filter out label-irrelevant patches and direct similar class-discriminative patches to the same expert. Experimental results on datasets such as MNIST, CIFAR-10, and CelebA support the theoretical claims, showing that pMoE can avoid learning spurious correlations.

PAD-Net: In this paper by He *et al*. [51] challenge conventional wisdom regarding dynamic networks by introducing a partially dynamic network called PAD-Net, which converts redundant dynamic parameters into static ones. They also develop Iterative Mode Partition to efficiently allocate dynamic and static parameters. The effectiveness of their approach is demonstrated through extensive experiments using two advanced dynamic architectures, DY-Conv and MoE, on image classification and GLUE benchmarks. Notably, PAD-Net achieves a 0.7% improvement in top-1 accuracy with only 30% dynamic parameters in ResNet-50, and a 1.9% increase in average score for language understanding with only 50% dynamic parameters in BERT, outperforming fully dynamic networks

StableMoE: In this paper by Dai [52] proposes a StableMoE to address the issue of routing fluctuation. This method involves two training phases. Initially, a balanced and cohesive routing strategy is being learned, which is then distilled into a separate lightweight router, detached from the backbone model. Subsequently, the distilled router is being employed to establish token-to-expert assignments, which are then fixed to ensure a stable routing strategy. Experimental validation is currently being conducted on tasks such as language modeling and multilingual machine translation, demonstrating that StableMoE surpasses previous MoE methods in terms of both convergence speed and performance.

Alpa: In this paper by Zheng *et al*. [36] proposes a parallel model training approach Alpa, a parallel model training approach that automates the process by generating execution plans integrating data, operator, and pipeline parallelism. Alpa addresses the challenge of scaling out complex DL models on distributed compute devices by distributing the training of large DL models and conceptualizing parallelisms into two hierarchical levels: inter-operator and intra-operator parallelisms. Building upon this framework, Alpa creates a new hierarchical space for extensive model-parallel execution plans. To automatically derive efficient parallel execution plans at each level of parallelism, Alpa employs a series of compilation passes. Additionally, Alpa implements an efficient runtime mechanism to coordinate the two-level parallel execution on distributed compute devices. Evaluation results indicate that Alpa produces parallelization plans that either match or surpass the performance of hand-tuned model-parallel training systems, even on models they are specifically designed for. Unlike specialized systems, Alpa also adapts to models with heterogeneous architectures and models without manually crafted plans.

BaGuaLu: In this paper by Ma *et al*. [53] presents the first endeavor aimed at training brain-scale models on an entire exascale supercomputer, specifically, the New Generation Sunway Supercomputer. BaGuaLu, achieved by integrating hardware-specific intra-node optimization with hybrid parallel strategies, exhibits commendable performance and scalability on remarkably large models. Evaluation results demonstrate that BaGuaLu can train models with 14.5 trillion parameters at a performance exceeding 1 EFLOPS using mixed precision. Moreover, it possesses the capability to train models with 174 trillion parameters, a scale comparable to the number of synapses in a human brain.

MEFT: In this paper by Hao *et al*. [54] introduces MEFT (Memory-Efficient Fine-Tuning), a method designed to fine-tune LLMs more efficiently by addressing the memory limitations often encountered in resource-constrained environments. MEFT leverages the sparsity in the activation of neurons within feed-forward networks (FFNs) and utilizes the larger capacity of CPU memory compared to GPU memory. By storing and updating larger adapter parameters on the CPU and using a mixture of experts (MoE)-like architecture, MEFT reduces unnecessary CPU computations and the communication volume between the CPU and GPU. The method dynamically retrieves and activates only relevant neurons for each input, significantly reducing GPU memory usage while maintaining fine-tuning performance. Experimental results demonstrate that MEFT achieves comparable results to traditional methods under resource-limited conditions, particularly in tasks requiring extensive knowledge, thereby proving its effectiveness in optimizing memory usage and computational efficiency.

GShard: In this paper by Lepikhin *et al*. [55] proposes conditional computation as a solution to the challenges mentioned above and illustrate its effectiveness and practicality. They extensively utilize GShard, a module comprising lightweight annotation APIs and an extension to the XLA compiler, to facilitate large-scale models with capacities of up to trillions of parameters. Leveraging GShard and conditional computation, they scale up a multilingual neural machine translation Transformer model with Sparsely-Gated Mixture-of-Experts. The authors demonstrate that such a massive model with 600 billion parameters can be efficiently trained on 2048 TPU v3 cores within 4 days, achieving significantly higher translation quality from 100 languages to English compared to previous approaches.

Parameter-efficient MoEs: In this paper by Zadouri *et al*. [56] proposes an extremely parameter-efficient architecture for Mixture-of-Experts (MoEs). This architecture leverages MoEs using lightweight settings, allowing for the fine-tuning of a dense model by updating less than 1% of its parameters. The instruction fine-tuning with the proposed methods consistently outperforms traditional parameter-efficient approaches on unseen tasks, while maintaining high parameter efficiency throughout different scales.

SMoE-Dropout: In this paper by Chen *et al*. [57] proposes a new plug-and-play training framework called SMoE-Dropout, designed to enhance the accuracy of transformers at full capacity without encountering collapse. SMoE-Dropout includes a randomly initialized and fixed router network to activate experts, gradually increasing the number of activated experts as training progresses. Transformers trained with SMoE-Dropout inherently display a "self-slimmable" property, adjusting to resource availability and delivering smooth and consistent performance improvements as the number of activated experts increases during inference or fine-tuning.

EdgeMoE: In this paper by Yi *et al*. [58] introduces EdgeMoE, the first on-device LLM inference engine capable of scaling the model size (number of experts) with both memory and time efficiency. The design of EdgeMoE is based on the unique observation that most computations are concentrated in a small portion of weights (non-experts are "hot weights") that can be stored in device memory, while most weights (experts are "cold weights") contribute minimally to computations. To enable an I/O-compute pipeline, EdgeMoE predicts which expert will be activated before executing its router function. It also introduces two novel techniques—expertwise bitwidth adaptation and in-memory expert management—to reduce the expert I/O overhead of EdgeMoE.

SE-MoE: In this paper by Shen *et al*. [59] proposes a framework capable of scaling Mixture-of-Experts (MoE) models to trillions

of parameters. It fully utilizes cluster resources, including HBM, CPU memory, and SSDs, to surpass memory limitations and achieve efficient training scheduling. Additionally, it employs 2D prefetch scheduling and fusion communication to enhance heterogeneous storage efficiency. A novel inference method based on ring memory uses dynamic graph scheduling to maximize the overlap of computation and communication, further enhancing inference performance for larger-scale MoE models without requiring additional machines.
ScheMoE: In this paper by Shi *et al*. [60] proposes an optimal scheduling framework for communication and computation tasks in training MoE models. ScheMoE integrates a novel all-to-all collective that efficiently utilizes both intra- and inter-connect bandwidths. It supports easy extensions of customized all-to-all collectives and data compression methods, all while benefiting from the proposed scheduling algorithm.
MoE-Mamba: In this paper by Pioro *et al*. [61] proposes a model that integrates Mamba with a Mixture of Experts (MoE) layer, called MoE-Mamba. This approach achieves the efficiency benefits of both SSMs and MoE, reaching the same performance as Mamba in 2.35× fewer training steps. The authors also confirm that the improvements achieved by MoE-Mamba are consistent across different model sizes, design choices, and the number of experts.
Pre-Gates MoE: In this paper by Hwang *et al*. [62] proposes the Pregated MoE system, which addresses the computational and memory issues of traditional MoE architectures through algorithm-system co-design. This system uses a novel pre-gating function to mitigate the dynamic aspect of sparse expert activation, enabling it to manage MoE's large memory footprint and deliver high performance. The Pre-gated MoE system is shown to enhance performance, decrease GPU memory usage, and maintain model quality. These characteristics make it possible to cost-effectively deploy large-scale LLMs using a single GPU with high performance.
NLLB:In this paper by Team NLLB [63] aims to break the 200 language barrier in machine translation while ensuring safe, high-quality results that consider ethical implications. The authors contextualize the need for low-resource language translation support through interviews with native speakers and create datasets and models to narrow the performance gap between low and high-resource languages. They develop a conditional compute model based on Sparsely Gated Mixture of Experts and use novel data mining techniques to obtain training data. The authors propose architectural and training improvements to counteract overfitting and evaluate the performance of over 40,000 translation directions using a human-translated benchmark, Flores-200. They also combine human evaluation with a novel toxicity benchmark to assess translation safety. The proposed model achieves a 44% relative improvement in BLEU score over the previous state-of-the-art, laying important groundwork towards a universal translation system.
EvoMoE: In this paper by Nie *et al*. [28] introduces EvoMoE, an efficient end-to-end training framework for Mixture-of-Experts (MoE) models. EvoMoE addresses the issues of immature experts and unstable sparse gates in existing MoE models by gradually evolving from a single expert to a large and sparse MoE structure. The framework consists of two phases: expert-diversify and gate-sparsify, and uses a novel Dense-to-Sparse gate (DTS-Gate) to route tokens to fewer experts. The authors evaluate EvoMoE on three popular models and tasks and show that it outperforms existing baselines.
FiLM: In this paper by Zhou *et al*. [64] present a novel approach to time series analysis and forecasting called the Frequency improved Legendre Memory (FiLM) model. This innovative architecture combines a mixture of experts for robust multiscale feature extraction with a redesigned Legendre Projection Unit (LPU), making it a versatile tool for data representation that addresses the challenge of preserving historical information. The model also incorporates Frequency Enhanced Layers (FEL), which leverage Fourier analysis and low-rank matrix approximation to reduce dimensionality, minimize noise, and mitigate overfitting in time series data. Through extensive experiments across six benchmark datasets spanning multiple domains, including energy, traffic, economics, weather, and disease, the authors demonstrate significant performance improvements. Their model outperforms state-of-the-art methods by 19.2% in multivariate forecasting and 26.1% in univariate forecasting. Furthermore, the dimensionality reduction achieved by FiLM yields substantial gains in computational efficiency. This research contributes valuable insights and tools to the field of time series analysis, offering enhanced accuracy and efficiency for forecasting tasks across diverse domains.
FastMoE: In this paper by He *et al*. [79] proposes a solution to the challenge of training trillion-scale language models. The authors highlight the significant potential of scaling language models to trillions of parameters but note that existing platforms for this task are limited to proprietary hardware and software stacks. FastMoE addresses this limitation by offering a PyTorch-based system compatible with common accelerators. It features a hierarchical interface for flexible model design and easy adaptation to various applications, along with highly optimized training speed through sophisticated acceleration techniques. Notably, FastMoE supports distributing experts across multiple GPUs and nodes, enabling linear scaling of expert numbers with available GPUs. This system aims to democratize large-scale MoE model training, making it more accessible to the broader research community, particularly those using GPUs and PyTorch, while providing the necessary performance for trillion-parameter models.
ACE: In this paper by Cai *et al*. [65] proposes a one-stage recognition approach called Ally Complementary Experts (ACE) for long-tailed datasets. In this approach, each expert specializes in a specific subset of data and is complementary to other experts in less frequently seen categories, without being affected by unseen data. To prevent overfitting, a distribution-adaptive optimizer is designed to adapt the learning pace of each expert.
HyperMoE: In this paper by Zhou *et al*. [66] proposes a new design for the Mixture of Experts (MoE) framework called HyperMoE, which includes a HyperExpert component. This novel method tackles a basic problem in MoE systems: striking a balance between sufficient expert availability and sparse expert selection. HyperMoE's higher performance above baselines based on Switch Transformers in a range of NLP tasks serves as evidence of its efficacy. The authors showcases a strong relationship between the experts who are finally picked and selection embeddings, which are formed from the environment of unselected experts. Their findings highlight the intricacy and effectiveness of the HyperMoE architecture in managing expert knowledge and selection inside the MoE framework by indicating that the selection embeddings successfully encode critical information about the knowledge required by the selected experts
BlackMamba: In this paper by Anthony *et al*. [67] proposes BlackMamba, a novel architecture that combines alternating attention-free Mamba blocks with routed MLPs. The authors design, implement, and evaluate this model, demonstrating its efficiency and effectiveness. They train and open-source two versions: 340M/1.5B BlackMamba and 630M/2.8B BlackMamba2. The study shows that BlackMamba requires significantly fewer training FLOPs to achieve comparable performance on downstream tasks compared to dense transformer models. Additionally, the paper explores the compounding inference benefits of combining attention-free architectures like

Mamba with routed sparsity architectures like MoE.

7. INTEGRATING MIXTURE OF EXPERTS INTO RECOMMENDATION ALGORITHMS

Many modern recommender systems leverage multi-task learning frameworks to create a more comprehensive understanding of user behavior. This approach simultaneously models multiple related objectives, such as user engagement (clicks, time spent), satisfaction (ratings, likes), and purchases. By jointly modeling these factors, systems can efficiently share knowledge and data across tasks, which is particularly beneficial for objectives with limited data availability. Additionally, multi-task learning serves as a regularizer, where auxiliary tasks introduce inductive biases that enhance the main task's generalization capabilities.Traditionally, many multi-task learning models in recommendation systems have relied on shared-bottom architectures. However, these structures often face challenges due to optimization conflicts arising from multiple tasks sharing the same parameters. Other common issues include data sparsity, heterogeneity, and the complexity of users' underlying intentions.To address these limitations, recent large-scale recommendation systems have begun adopting the Multi-gate Mixture-of-Experts (MMoE) model for multi-task learning. This approach has yielded state-of-the-art results in recommendations by allowing for more flexible and task-specific parameter allocation. The following section will highlight notable industry examples of MMoE implementation in recommender systems, demonstrating its effectiveness in overcoming the challenges associated with traditional multi-task learning approaches.

MoME:In this paper by Xu *et al*. [68] introduces a novel approach to multi-task learning in recommendation systems. MoME builds upon the MoE architecture by incorporating a masking mechanism to enhance efficiency and performance. The authors address the challenges of computational complexity and negative transfer in traditional MoE models by dynamically selecting a subset of experts for each task. This selective activation is achieved through learnable binary masks, which determine which experts contribute to each task's prediction. The paper demonstrates that MoME significantly reduces computational costs while maintaining or improving performance across various recommendation tasks. Furthermore, the authors show that their approach mitigates negative transfer between tasks by allowing for more specialized expert utilization. Extensive experiments on both public and industrial datasets validate the effectiveness of MoME, showcasing its ability to outperform existing multi-task recommendation models in terms of both efficiency and accuracy. The paper also provides insights into the interpretability of expert assignments across different tasks, offering a deeper understanding of the model's decision-making process.

SummaReranker: In this paper by Ravaut *et al*. [69] proposes that while sequence-to-sequence neural networks, especially those based on large pre-trained language models fine-tuned for specific tasks, have significantly improved abstractive summarization, they still face challenges. These models typically rely on beam search to produce a single summary from numerous possibilities. However, this decoding method is not ideal due to exposure bias. In this research, the author demonstrates that it's possible to train an additional model to re-evaluate and rank a set of potential summaries. The proposed solution, called SummaReranker, utilizes a mixture-of-experts architecture. This approach is designed to identify the most suitable summary candidate and consistently improves upon the performance of the original model.

PLE: In this paper by Tang *et al*. [70] addressing key challenges in multi-task learning (MTL) for recommendation systems. They tackle performance degeneration due to negative transfer and the "seesaw phenomenon" where improving one task's performance often degrades others. The PLE model achieves this through explicit separation of shared and task-specific components, coupled with a progressive routing mechanism for gradual extraction and separation of deeper semantic knowledge. The authors rigorously test PLE on a massive Tencent video recommendation dataset, conduct online evaluations on a large-scale content recommendation platform, and perform experiments on public benchmark datasets across various scenarios. Results consistently show PLE outperforming state-of-the-art MTL models, with significant online improvements in view-count and watch time. Notably, PLE successfully eliminates the seesaw phenomenon across diverse applications. The model's effectiveness and practical value are further validated by its successful deployment in Tencent's online video recommender system, marking a significant advancement in MTL for large-scale recommendation environments.

MDFEND: In this paper by Nan *et al*. [71] proposes the challenge of multi-domain fake news detection (MFND), an emerging field that aims to improve upon single-domain approaches. The authors identify domain shift as a major obstacle, where varying data distributions across different domains hinder the effectiveness of existing fake news detection techniques. To tackle this issue, they introduce two key contributions. First, they create Weibo21, a benchmark dataset for MFND containing 4,488 fake news items and 4,640 real news items across 9 domains, each with domain labels. Second, they propose MDFEND (Multi-domain Fake News Detection Model), which employs a domain gate to aggregate multiple representations extracted by a mixture of experts. Experimental results demonstrate that MDFEND significantly enhances multi-domain fake news detection performance compared to existing methods.

CAME: In this paper by Guo *et al*. [72] proposes a novel approach for competitive learning that fosters the growth and improvement of each expert's area of expertise in certain relevance patterns. The author shows that their model, CAME, considerably outperforms state-of-the-art baselines in both in-domain and out-of-domain contexts through comprehensive testing on a variety of retrieval benchmarks.

8. PYTHON LIBRARIES FOR MOE

MoE-Infinity:In this paper by Xue *et al*. [80] proposes MoE-Infinity, a library for MoE inference and serving that is cost-effective, fast, and user-friendly. The library employs sequence-level expert activation tracing, a novel method that excels at detecting sparse activations and exploiting the temporal locality of MoE inference. By examining these traces, MoE-Infinity implements activation-aware expert prefetching and caching, which significantly lowers the latency overheads typically associated with offloading experts, resulting in enhanced cost performance.

SMT 2.0: In this paper by Saves *et al*. [73] introduces a freely available Python library that provides a comprehensive suite of surrogate modeling approaches, sampling strategies, and benchmark problems. This open-source tool is designed to support researchers and practitioners in the field of surrogate modeling and optimization.SMT 2.0 is the first open-source surrogate library to propose surrogate models for hierarchical and mixed inputs.

9. CONCLUSION

Theoretically, a deeper understanding of MoE architectures and their working principles is needed. As we saw in [74] paper, the reasons

behind the success of MoE layers are still partially obscure. Therefore, more theoretical and empirical research is required to demystify the intrinsic mechanics of these models, potentially leading to their optimization and better generalization.Additionally, how to design more effective gating mechanisms and expert models is an open question with great potential for future exploration. While Expert Choice Routing offers a promising direction, other innovative approaches might enhance the routing mechanism.Lastly, while MoE has shown impressive results in domains like NLP and computer vision, there is considerable room to explore its utility in other domains, such as reinforcement learning, tabular data domains, and more.The journey of MoE is in its infancy in the realm of deep learning, with many milestones yet to be achieved. However, its potential for transforming how we understand and deploy deep learning models is enormous. With the current state of computing, it's unlikely that we will see significant improvements to hardware as rapidly as we see improvements to modeling techniques. By leveraging the inherent strength of the MoE paradigm—the division of complex tasks into simpler subtasks handled by specialized expert models—we may continue to push the boundaries of what is achievable with deep learning. And that, indeed, is an exciting prospect to look forward to.

10. REFERENCES

[1] Robert A. Jacobs, Michael I. Jordan, and Steven Nowlan, "Adaptive Mixtures of Local Experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 03 1991.

[2] Moe Nandi Aung, Yati Phyo, Canh Minh Do, and Kazuhiro Ogata, "A divide and conquer approach to eventual model checking," *Mathematics*, vol. 9, no. 4, 2021.

[3] 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," 2017.

[4] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever, "Learning factored representations in a deep mixture of experts," 2014.

[5] 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," 2021.

[6] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1930–1939.

[7] Hanxue Liang, Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, and Zhangyang Wang, "M3vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design," *ArXiv*, vol. abs/2210.14793, 2022.

[8] Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu, "Mova: Adapting mixture of vision experts to multimodal context," 2024.

[9] Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen, "Moebert: from bert to mixture-of-experts via importance-guided adaptation," in *North American Chapter of the Association for Computational Linguistics*, 2022.

[10] Chenqi Kong, Anwei Luo, Peijun Bao, Yi Yu, Haoliang Li, Zengwei Zheng, Shiqi Wang, and Alex C. Kot, "Moe-ffd: Mixture of experts for generalized and parameter-efficient face forgery detection," 2024.

[11] Alex Jinpeng Wang, Linjie Li, Yiqi Lin, Min Li, Lijuan Wang, and Mike Zheng Shou, "Leveraging visual tokens for extended text contexts in multi-modal learning," 2024.

[12] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, and Furu Wei, "Vlmo: Unified vision-language pre-training with mixture-of-modality-experts," 2022.

[13] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Jinfa Huang, Junwu Zhang, Munan Ning, and Li Yuan, "Moe-llava: Mixture of experts for large vision-language models," 2024.

[14] 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 *Proceedings of the 39th International Conference on Machine Learning*, 2022, pp. 18332–18346.

[15] Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi, "Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, Eds. 2021, vol. 34, pp. 29335–29347, Curran Associates, Inc.

[16] Chongjian Ge, Junsong Chen, Enze Xie, Zhongdao Wang, Lanqing Hong, Huchuan Lu, Zhenguo Li, and Ping Luo, "Metabev: Solving sensor failures for 3d detection and map segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 8721–8731.

[17] Tianlong Chen, Xuxi Chen, Xianzhi Du, Abdullah Rashwan, Fan Yang, Huizhong Chen, Zhangyang Wang, and Yeqing Li, "Adamv-moe: Adaptive multi-task vision mixture-of-experts," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17300–17311, 2023.

[18] Zhida Feng, Zhenyu Zhang, Xintong Yu, Yewei Fang, Lanxin Li, Xuyi Chen, Yuxiang Lu, Jiaxiang Liu, Weichong Yin, Shikun Feng, Yu Sun, Li Chen, Hao Tian, Hua Wu, and Haifeng Wang, "Ernie-vilg 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 10135–10145.

[19] Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang, "Loramoe: Alleviate world knowledge forgetting in large language models via moe-style plugin," 2024.

[20] Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui, "Glam: Efficient scaling of language models with mixture-of-experts," 2022.

[21] Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou, "Mixture-of-experts meets instruction tuning:a winning combination for large language models," 2023.

[22] Zeyue Xue, Guanglu Song, Qiushan Guo, Boxiao Liu, Zhuofan Zong, Yu Liu, and Ping Luo, "Raphael: Text-to-image generation via large mixture of diffusion paths," 2024.

[23] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed, "Mixtral of experts," 2024.

[24] Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-Wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen, "Cumo: Scaling multimodal llm with co-upcycled mixture-of-experts," 2024.

[25] Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Rozière, Jacob Kahn, Daniel Li, Wen tau Yih, Jason

Weston, and Xian Li, "Branch-train-mix: Mixing expert llms into a mixture-of-experts llm," 2024.

[26] Junmo Kang, Leonid Karlinsky, Hongyin Luo, Zhen Wang, Jacob Hansen, James Glass, David Cox, Rameswar Panda, Rogerio Feris, and Alan Ritter, "Self-moe: Towards compositional large language models with self-specialized experts," 2024.

[27] Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng, "When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications," 2024.

[28] Xiaonan Nie, Pinxue Zhao, Xupeng Miao, Tong Zhao, and Bin Cui, "Hetumoe: An efficient trillion-scale mixture-of-expert distributed training system," 2022.

[29] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu, "ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers," 2023.

[30] Jinjin Xu, Liwu Xu, Yuzhe Yang, Xiang Li, Fanyi Wang, Yanchun Xie, Yi-Jie Huang, and Yaqian Li, "u-llava: Unifying multi-modal tasks via large language model," 2024.

[31] Elias Frantar and Dan Alistarh, "Qmoe: Practical sub-1-bit compression of trillion-parameter models," 2023.

[32] Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, Joe Chau, Peng Cheng, Fan Yang, Mao Yang, and Yongqiang Xiong, "Tutel: Adaptive mixture-of-experts at scale," *ArXiv*, vol. abs/2206.03382, 2022.

[33] Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia, "Megablocks: Efficient sparse training with mixture-of-experts," 2022.

[34] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham, "Jamba: A hybrid transformer-mamba language model," 2024.

[35] Zhenxing Mi and Dan Xu, "Switch-nerf: Learning scene decomposition with mixture of experts for large-scale neural radiance fields," in *International Conference on Learning Representations*, 2023.

[36] Lianmin Zheng, Zhuohan Li, Hao Zhang, Yonghao Zhuang, Zhifeng Chen, Yanping Huang, Yida Wang, Yuanzhong Xu, Danyang Zhuo, Eric P. Xing, Joseph E. Gonzalez, and Ion Stoica, "Alpa: Automating inter- and intra-operator parallelism for distributed deep learning," 2022.

[37] 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," 2024.

[38] Zhao You, Shulin Feng, Dan Su, and Dong Yu, "Speechmoe: Scaling to large acoustic models with dynamic routing mixture of experts," 2021.

[39] Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang, "Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models," 2024.

[40] Xun Wu, Shaohan Huang, Wenhui Wang, and Furu Wei, "Multi-head mixture-of-experts," 2024.

[41] Yikang Shen, Zhen Guo, Tianle Cai, and Zengyi Qin, "Jetmoe: Reaching llama2 performance with 0.1m dollars," 2024.

[42] Xun Wu, Shaohan Huang, and Furu Wei, "Mixture of lora experts," 2024.

[43] Zexuan Zhong, Mengzhou Xia, Danqi Chen, and Mike Lewis, "Lory: Fully differentiable mixture-of-experts for autoregressive language model pre-training," 2024.

[44] 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," 2024.

[45] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia, "Mini-gemini: Mining the potential of multi-modality vision language models," 2024.

[46] Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby, "From sparse to soft mixtures of experts," 2023.

[47] 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," 2022.

[48] Jinguo Zhu, Xizhou Zhu, Wenhai Wang, Xiaohua Wang, Hongsheng Li, Xiaogang Wang, and Jifeng Dai, "Uni-perceiver-moe: Learning sparse generalist models with conditional moes," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds. 2022, vol. 35, pp. 2664–2678, Curran Associates, Inc.

[49] Junyang Lin, An Yang, Jinze Bai, Chang Zhou, Le Jiang, Xianyan Jia, Ang Wang, Jie Zhang, Yong Li, Wei Lin, Jingren Zhou, and Hongxia Yang, "M6-10t: A sharing-delinking paradigm for efficient multi-trillion parameter pretraining," 2021.

[50] Mohammed Nowaz Rabbani Chowdhury, Shuai Zhang, Meng Wang, Sijia Liu, and Pin-Yu Chen, "Patch-level routing in mixture-of-experts is provably sample-efficient for convolutional neural networks," 2023.

[51] Shwai He, Liang Ding, Daize Dong, Boan Liu, Fuqiang Yu, and Dacheng Tao, "Pad-net: An efficient framework for dynamic networks," 2023.

[52] Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei, "Stablemoe: Stable routing strategy for mixture of experts," 2022.

[53] Zixuan Ma and He, "Bagualu: targeting brain scale pretrained models with over 37 million cores," New York, NY, USA, 2022, PPoPP '22, p. 192–204, Association for Computing Machinery.

[54] Jitai Hao, WeiWei Sun, Xin Xin, Qi Meng, Zhumin Chen, Pengjie Ren, and Zhaochun Ren, "Meft: Memory-efficient fine-tuning through sparse adapter," 2024.

[55] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen, "{GS}hard: Scaling giant models with conditional computation and automatic sharding," in *International Conference on Learning Representations*, 2021.

[56] 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," 2023.

[57] Tianlong Chen, Zhenyu (Allen) Zhang, Ajay Jaiswal, Shiwei Liu, and Zhangyang Wang, "Sparse moe as the new dropout: Scaling dense and self-slimmable transformers," *ArXiv*, vol. abs/2303.01610, 2023.

[58] Rongjie Yi, Liwei Guo, Shiyun Wei, Ao Zhou, Shangguang Wang, and Mengwei Xu, "Edgemoe: Fast on-device inference of moe-based large language models," *ArXiv*, vol. abs/2308.14352, 2023.

[59] Liang Shen, Zhihua Wu, Weibao Gong, Hongxiang Hao, Yangfan Bai, Huachao Wu, Xinxuan Wu, Haoyi Xiong, Dianhai Yu, and Yanjun Ma, "Se-moe: A scalable and efficient mixture-of-experts distributed training and inference system," *ArXiv*, vol. abs/2205.10034, 2022.

[60] Shaohuai Shi, Xinglin Pan, Qiang Wang, Chengjian Liu, Xiaozhe Ren, Zhongzhe Hu, Yu Yang, Bo Li, and Xiaowen Chu, "Schemoe: An extensible mixture-of-experts distributed training system with tasks scheduling," *Proceedings of the Nineteenth European Conference on Computer Systems*, 2024.

[61] Maciej Pi'oro, Kamil Ciebiera, Krystian Kr'ol, Jan Ludziejewski, and Sebastian Jaszczur, "Moe-mamba: Efficient selective state space models with mixture of experts," *ArXiv*, vol. abs/2401.04081, 2024.

[62] Ranggi Hwang, Jianyu Wei, Shijie Cao, Changho Hwang, Xiaohu Tang, Ting Cao, Mao Yang, and Minsoo Rhu, "Pre-gated moe: An algorithm-system co-design for fast and scalable mixture-of-expert inference," *ArXiv*, vol. abs/2308.12066, 2023.

[63] NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, and Maha Elbayad, "No language left behind: Scaling human-centered machine translation," 2022.

[64] Tian Zhou, Ziqing MA, xue wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin, "Film: Frequency improved legendre memory model for long-term time series forecasting," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds. 2022, vol. 35, pp. 12677–12690, Curran Associates, Inc.

[65] Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang, "Ace: Ally complementary experts for solving long-tailed recognition in one-shot," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 112–121.

[66] Hao Zhao, Zihan Qiu, Huijia Wu, Zili Wang, Zhaofeng He, and Jie Fu, "Hypermoe: Towards better mixture of experts via transferring among experts," 2024.

[67] Quentin Anthony, Yury Tokpanov, Paolo Glorioso, and Beren Millidge, "Blackmamba: Mixture of experts for state-space models," 2024.

[68] Jiahui Xu, Lu Sun, and Dengji Zhao, "Mome: Mixture-of-masked-experts for efficient multi-task recommendation," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2024, SIGIR '24, p. 2527–2531, Association for Computing Machinery.

[69] Mathieu Ravaut, Shafiq Joty, and Nancy F. Chen, "Summareranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization," 2023.

[70] 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*, New York, NY, USA, 2020, RecSys '20, p. 269–278, Association for Computing Machinery.

[71] Qiong Nan, Juan Cao, Yongchun Zhu, Yanyan Wang, and Jintao Li, "Mdfend: Multi-domain fake news detection," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, New York, NY, USA, 2021, CIKM '21, p. 3343–3347, Association for Computing Machinery.

[72] Jiafeng Guo, Yinqiong Cai, Keping Bi, Yixing Fan, Wei Chen, Ruqing Zhang, and Xueqi Cheng, "Came: Competitively learning a mixture-of-experts model for first-stage retrieval," jul 2024.

[73] Paul Saves, Rémi Lafage, Nathalie Bartoli, Youssef Diouane, Jasper Bussemaker, Thierry Lefebvre, John T. Hwang, Joseph Morlier, and Joaquim R.R.A. Martins, "Smt 2.0: A surrogate modeling toolbox with a focus on hierarchical and mixed variables gaussian processes," *Advances in Engineering Software*.

[74] Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li, "Towards understanding mixture of experts in deep learning," 2022.

[75] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby, "Multimodal contrastive learning with limoe: the language-image mixture of experts," *ArXiv*, vol. abs/2206.02770, 2022.

[76] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen, "Lora: Low-rank adaptation of large language models," 2021.

[77] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, and Anthony Hartshorn, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[78] Ze-Feng Gao, Peiyu Liu, Wayne Xin Zhao, Zhong-Yi Lu, and Ji-Rong Wen, "Parameter-efficient mixture-of-experts architecture for pre-trained language models," 2022.

[79] Jiaao He, Jiezhong Qiu, Aohan Zeng, Zhilin Yang, Jidong Zhai, and Jie Tang, "Fastmoe: A fast mixture-of-expert training system," 2021.

[80] Leyang Xue, "Moe-infinity: Activation-aware expert offloading for efficient moe serving," in *https://arxiv.org/abs/2401.14361*, 2024.