Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 5, No. 2, 2025

The Rise of Sparse Mixture-of-Experts: A Survey from Algorithmic Foundations to Decentralized Architectures and Vertical Domain Applications

Dong Pan [1], Bingtao Li [1, *], Yongsheng Zheng [1], Jiren Ma [1], Victor Fei [2]

[1] Fedimoss Tech Hk Limited, Hong Kong, China

[2] Ormi Labs, Inc., California, USA

[*] **Corresponding author**: Bingtao Li (Email: bingtaoli@fedimoss.com)

Abstract: The sparse Mixture of Experts (MoE) architecture has evolved as a powerful approach for scaling deep learning models to more parameters with comparable computation cost. As an important branch of large language model (LLM), MoE model only activate a subset of experts based on a routing network. This sparse conditional computation mechanism significantly improves computational efficiency, paving a promising path for greater scalability and cost-efficiency. It not only enhance downstream applications such as natural language processing, computer vision, and multimodal in various horizontal domains, but also exhibit broad applicability across vertical domains including medical diagnosis, autonomous driving, financial analysis, and business intelligence. Despite the growing popularity and application of MoE models across various domains, there lacks a systematic exploration of recent advancements of MoE in many important fields. Existing surveys on MoE suffer from limitations such as lack coverage or not extensively exploration of key areas. This survey seeks to fill these gaps. In this paper, First ly, we examine the foundational principles of MoE, with an in-depth exploration of its core components—the routing network and expert network. Subsequently, we extend beyond the centralized paradigm to the decentralized paradigm, which unlocks the immense untapped potential of decentralized infrastructure, enables democratization of MoE development for broader communities, and delivers greater scalability and cost-efficiency. Furthermore we focus on exploring its vertical domain applications. Finally, we also identify key challenges and promising future research directions. To the best of our knowledge, this survey is currently the most comprehensive review in the field of MoE. We aim for this article to serve as a valuable re source for both researchers and practitioners, enabling them to navigate and stay up-to-date with the latest advancements.

Keywords: Mixture-of-Experts, Decentralized Learning, LLM, Transformer.

1. Introduction

The Recent advances in artificial intelligence (AI), especially regarding large language model (LLM), predominantly stem from scaling principles [2, 11, 43] that larger model sizes bring better model quality—a phenomenon formally described as the Scaling Law. Despite its oversimplified nature, this fundamental principle continues to steer AI research evolution. However, extreme-scale model expansion also incurs extremely high computational costs.

To this end, architectures based on sparse Mixture of Experts (MoE) [16, 20, 24, 47, 84, 129] have paved a promising path, enabling the scaling of foundation models to larger sizes at comparable computational cost. A recent open-source MoE model [54] which integrates sparsely-gated MoE with Transformer-based foundation models [76, 97], has surpassed other open-source alternatives and demonstrated performance comparable to prominent closed-source models such as GPT-4o [35], thereby unlocking the broader application potential of this three-decade-old technology [37].

Beyond efficiency advantages, the MoE architectures also offer opportunities to enhance model interpretability [5, 46, 66, 70, 129]. By learning its intrinsic allocation mechanism, researchers can gain insights into how different "experts" specialize in handling specific types of data or tasks. This interpretability not only deepens our understanding of model behavior but also paves new pathways for designing more robust and transparent AI systems.

However, some recent MoE models [92, 116] have scaled to 1T parameters with context windows exceeding 128K tokens, dramatically increasing computational resource demands. This exponential growth poses significant challenges. High-performance computing clusters for advanced MoE research and development remain unaffordable for resource-limited individual researchers and small laboratories. Only a handful of large corporations and institutions possess sufficient resources to develop such models, creating quasi-monopolies that stifle AI innovation.

Urgent adoption of efficient training and inference paradigms is imperative for sustainable scaling. Current advanced frameworks [26, 36, 69, 77, 115, 127] primarily utilize limited homogeneous resources, operating under centralized paradigm. Crucially, decentralized clusters and consumer-level devices, usually have a significant amount of computing resources than centralized clusters. Despite harboring immense untapped potential, these computing resources are typically overlooked due to lower bandwidth and compute power. Decentralized paradigm [60, 82] emerges as a promising solution. This distributed paradigm integrates heterogeneous resources across individual consumer GPUs, and clusters, fully utilizes the computing resources, enables democratization of MoE development for broader communities, delivers greater scalability and cost-efficiency.

1.1. Related Work

While several related surveys predate this work, notable gaps still remain. For instance, among the comprehensive

studies, [122] covers pre-deeplearning developments, [23] omits recent breakthroughs in the field, [12, 65] lack coverage of decentralized MoE paradigms, and none extensively explore MoE applications in vertical industries. Furthermore, there are focus-limited studies, [27] focuses on big data applications, [55] focuses on inference acceleration, [114] focuses on applications of wireless communication scenarios. Our survey bridges these gaps by conducting an in-depth exploration of MoE architectures in both centralized and decentralized infrastructure, scenario-specific applications in critical vertical domains. Table 1 highlights the differences.

Table 1. Comparison of our survey with related surveys

Surveys	[122]	[23]	[12]	[65]	[27]	[55]	[114]	Ours
Comprehensive introduction of MoE core designs and recent advancements	×	×	✓	✓	×	×	✓	✓
Delineation of the decentralized architecture paradigm	×	×	×	×	×	×	×	✓
Extensively exploration of vertical domain applications	×	×	×	×	✓	×	×	✓

1.2. Contributions

Comprehensive and Timely Survey: We present the most comprehensive review in the field of MoE, systematically identifying studies from algorithmic foundations to decentralized architectures and vertical domain applications. Our survey meticulously analyzes relevant research, examining their motivations, technical principles, and key factors requiring consideration within them, offering a valuable reference for researchers and practitioners, tracking the latest research developments and inspiring new ideas in this explosively evolving field.

In-Depth Exploration of Decentralized Architecture Paradigm: We delve into decentralized paradigm, which unlocks the immense untapped potential of decentralized infrastructure. Our survey delineates critical challenges faced by the decentralized paradigm and reviews existing related research efforts addressing these issues, moving beyond traditional centralized paradigm toward enabling democratization of MoE development for broader communities, paving the way for greater scalability and cost-efficiency. This evolves the development paradigm for MoE models from "competing for resources" to "on-demand scalability."

Diverse Applications of Vertical Domains: We extensively explore applications of MoE in typical vertical domains, providing an overall understanding of how MoE can be applied to specific tasks.

The remainder of this survey is organized as follows: Section 2 comprehensively introduces the core component design of sparse MoE models and examines various factors requiring consideration in these components. Section 3 extends beyond the centralized paradigm to delve into the decentralized paradigm, unlocking their potential for collaborative training and inference of MoE models in heterogeneous environments. Section 4 shifts focus from widely studied horizontal applications (e.g., NLP, CV, multimodal) to vertical industry applications, such as medical diagnosis, autonomous driving, finance analysis, business intelligence and blockchain. This aims to provide readers with a comprehensive reference for implementing MoE in specific tasks. Section 5 analyzes critical challenges and emerging opportunities. Section 6 concludes this paper. The overall structure of this survey is illustrated in Figure 1:

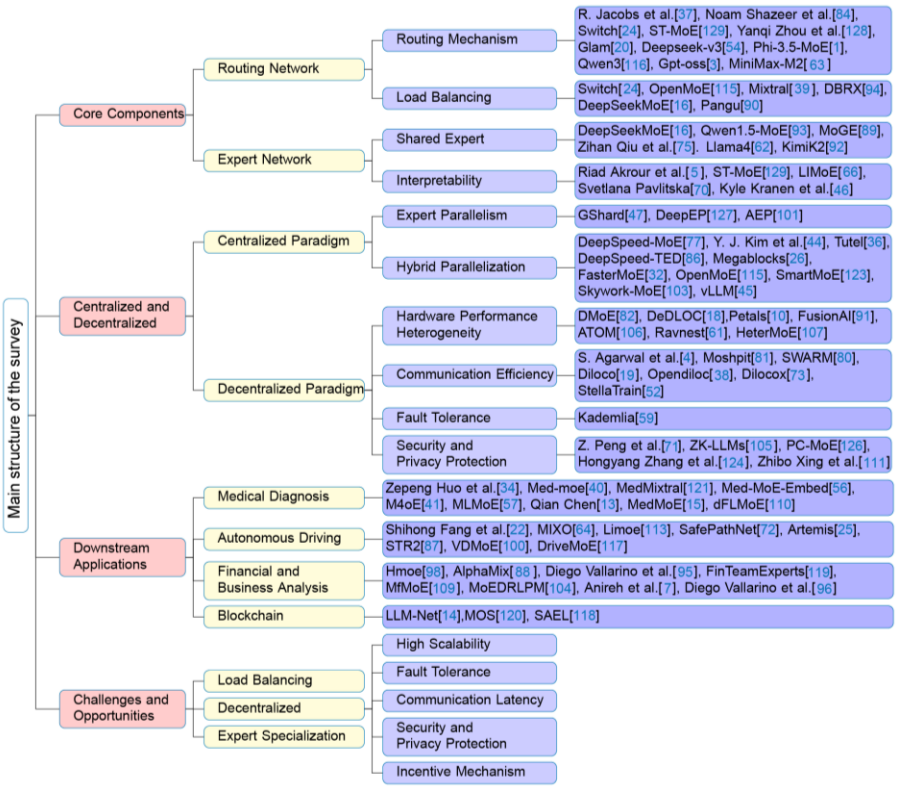

Figure 1. Main structure of the survey

2. Core Components of MoE

The concept of MoE originated from the 1991 Adaptive Mixture of Local Experts [37]. MoE model introduces the idea of experts, dividing the network into multiple independent subnetworks (called experts). MoE model dynamically determines which experts should process which tokens, typically activating only the most relevant few experts for each token. The remaining experts do not participate in the computation, significantly improving computational efficiency. This allows for a significant increase in model parameter size under the same computational budget [17, 23, 122]. Table 2 summarizes the key characteristics of state-of-the-art open-source MoE models.

The MoE model primarily consists of two key components:

Routing Network: This component determines which tokens are sent to which experts. The routing mechanism is a critical aspect of MoE systems. Routing network is composed of learnable parameters and is trained alongside the rest of the network during pre-training.

Expert Network: By splitting the dense feed-forward neural network (FFN) layers into multiple independent parts, each of which is an independent neural network—expert network. In practice, these experts are typically FFN units, but they can also be CNN or more complex network structures.

Table 2. A Summary of State-of-the-Art Open-Source MoE Models

Model	Affiliation	Expert Count (Routed + Shared)	Activation Count (Routed + Shared)	Model Params (Activation/Total)	Time
DeepSeekMoE [16]	DeepSeek	64+2	6+2	2.8B/16.4B	2024.01
DBRX [94]	Databricks	16+0	4+0	36B/132B	2024.03
Mixtral [39]	Mistral AI	8+0	2+0	39B/141B	2024.04
Phi-3.5-MoE [1]	Microsoft	16+0	2+0	6.6B/42B	2024.08
Deepseek-v3 [54]	DeepSeek	256+1	8+1	37B/671B	2024.12
Llama4 [62]	Meta	16+1 128+1	1+1 1+1	17B/109B 17B/400B	2025.04
Qwen3 [116]	Alibaba	128+0 128+0	8+0 8+0	3B/30B 22B/235B	2025.05
Pangu Ultra MoE [90]	Huawei	256+1	8+1	39B/718B	2025.05
KimiK2 [92]	Moonshot	384+1	8+1	32B/1T	2025.07
Gpt-oss [3]	OpenAI	32+0 128+0	4+0 4+0	3.6B/21B 5.1B/117B	2025.08
Qwen3-Next [116]	Alibaba	512+1	10+1	3B/80B	2025.09
MiniMax-M2 [63]	MiniMax	256+0	8+0	10B/229B	2025.10

In the following section, we will delve into the core components of the MoE model, introduce the basic principles of key algorithms, and discuss various factors that need to be considered within them.

2.1. Routing Network

The routing network, also called the router, which is used to determine which tokens are route to which experts. The sparse gating functions activate a selected subset of experts or tokens, which can be considered as a form of conditional computation.

The routing mechanism have been studied extensively [20, 24, 47, 54, 84, 116]. In this section, we provide an in-depth review of two common categories: (1) token choice routing, where each token selects the best top-k experts, and (2) expert choice routing, where each expert picks the top-k tokens. [84] introduces a token choice routing mechanism. In this routing mechanism, only the top-k experts determined by the router are forward-passed, while the computations of the other experts are skipped, achieving sparse activation of experts, as shown in Figure 2. The computation process is shown in the formula below:

$$\mathrm{MoE}(x) = \sum_{i=1}^{n}(\mathrm{G}_i(\mathrm{x})\mathrm{E}_i(\mathrm{x})) \quad (1)$$

$$G(\mathbf{x}) = \mathrm{Softmax}(\mathrm{TopK}(H(x), k)) \quad (2)$$

$$H(\mathbf{x}) = (\mathbf{x}.W_g)_i + StandardNormal().Softplus((\mathbf{x}.W_{\mathrm{noise}})_i) \quad (3)$$

$$\mathrm{TopK}(v,k) = \begin{cases} v_i, & if\ v_i\ belongs\ to\ the\ top\ k\ of\ v, \\ -\infty, & otherwise. \end{cases} \quad (4)$$

Where the router G(x) is obtained by multiplying the input x with a learnable weight matrix and Gaussian noise, followed by applying the Softmax function, k represents the number of experts activated per token. TopK(v, k) refers to selecting the top k largest elements from the vector v, while assign the rest to $-\infty$. The Softmax output will result in probabilities of 0 for these $-\infty$ elements. Finally, only the top-k experts are selected for each token.

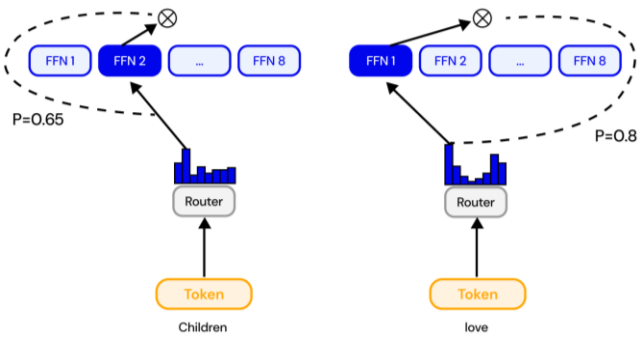

Figure 2. Illustration of token choice routing

The token choice routing mechanism allows each token to select the top-k experts. However, these independent token-level decisions often lead to expert load imbalance, resulting in reduced training efficiency and suboptimal model convergence. Despite previous works add an auxiliary loss on load balancing to mitigate these issues, this auxiliary loss does not guarantee a balanced load and may potentially degrade model performance.

Proposes a new routing method for sparsely activated MoE

model, termed expert choice routing [128]. The key idea of the routing method is to have experts pick the top-k tokens rather than tokens selecting top-k experts, as shown in Figure 3. Crucially, it allows tokens to be processed by varying numbers of experts while achieving perfect load balancing. This design simultaneously resolves two key limitations: (1) the expert-token assignment imbalance that leaves some experts under-optimized, and (2) the uniform expert count per token that ignores varying task relevance across tokens. Consequently, it enhances both training efficiency and downstream task performance.

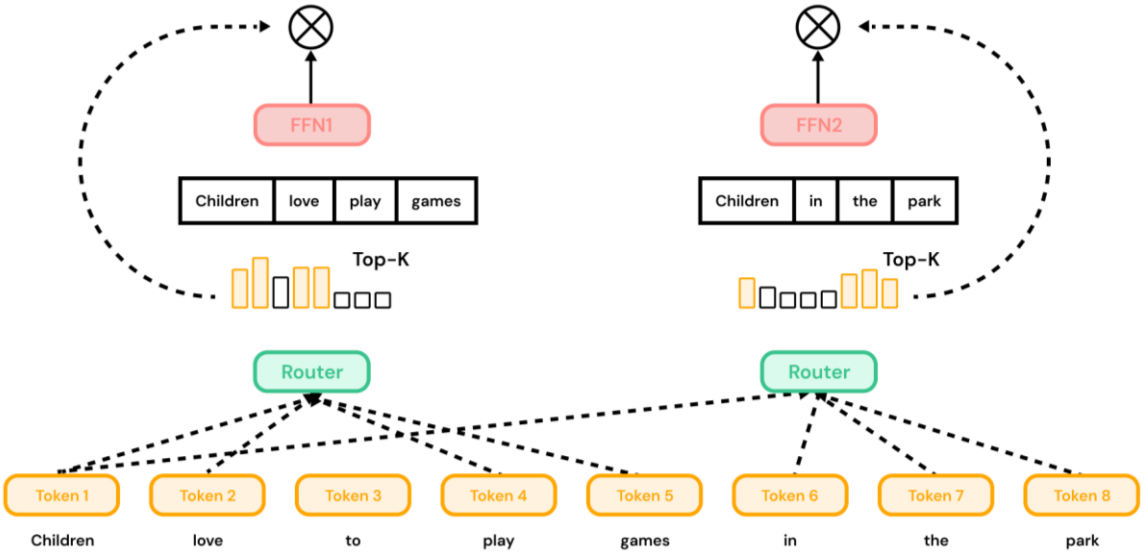

Figure 3. Illustration of expert choice routing

2.1.1. Load Balancing

In sparse MoE models, load balancing is a critical issue. Since only a small number of experts are activated, some experts may become overloaded with too many tokens, while others remain underutilized. This imbalance can lead to wasted computational resources and even cause routing collapse, ie., all tokens are routed to a few experts, leaving others undertrained and ineffective. If experts are distributed across multiple devices, load imbalance can exacerbate computational bottlenecks.

Auxiliary Loss: To mitigate this issue, Expert-Level Balance Loss and Device-Level Balance Loss as auxiliary losses [24, 39, 47, 90, 115] are added to the total model loss. These losses encourage tokens to be evenly distributed among experts, addressing expert utilization imbalance and cross-device computational load imbalance, respectively. However, this strategy may negatively impact model performance, requiring tuning the hyperparameter α to control the proportion of auxiliary loss in the total model loss, trading off load balancing and performance. The model's total training loss function is defined as:

$$L_{\text{total}} = L_{moe} + \alpha L_{aux} \tag{5}$$

Where L_{moe} represents the loss function of the standard MoE, such as cross-entropy loss. In the calculation formula of L_{total}, the hyperparameter α is used to control the weight of the auxiliary loss in the overall model loss. Notably, a large value degrades the model's performance, while a small value leads to training instability and resource underutilization.

As previously mentioned, if all tokens are sent to only a few popular experts, the training efficiency will decrease. In standard MoE training, the routing network tends to primarily activate the same few experts, a tendency that can become self-reinforcing, as popular experts train faster and thus are more likely to be selected. To alleviate this issue, an expert-level auxiliary loss [24, 47] is introduced to encourage assigning equal importance to all experts. This loss ensures that all experts receive approximately the same number of training samples, thereby balancing the selection among experts, as detailed below:

$$P_i = \frac{1}{T}\sum_{t=1}^{T} s_{i,t} \tag{6}$$

$$f_i = \frac{1}{K_r T}\sum_{t=1}^{T} 1\left(s_{i,t} \in Topk\left(s_{j,t} | 1 \le j \le N_r, K_r\right)\right) \tag{7}$$

$$L_{aux} = \alpha N_r \sum_{i=1}^{N_r} f_i P_i \tag{8}$$

Where $s_{i,t}$ denotes the routing probability of expert i for token t forward the gating network. P_i represents the average routing probability of expert i on a set of T input tokens. If the N_r experts are load-balanced, then the P_i for each expert should be $\frac{1}{Nr}$. fi represents the probability that an arbitrary token is routed to expert i. If the N_r experts are load-balanced, then the f_i for each expert should be $\frac{1}{N_r}$.

For the auxiliary loss L_{aux}, the larger the differences in probabilities across experts, the larger the value of L_{aux}. Conversely, the smaller the differences, the smaller L_{aux} becomes. The minimum value is achieved only when $f_i = \frac{1}{N_r}$ and $P_i = \frac{1}{N_r}$, indicating perfect load balancing.

In a distributed environment, MoE models distribute experts across multiple devices. If certain devices frequently activate experts while others rarely use their experts, it can lead to imbalanced computation load, affecting training efficiency. [16] further introduces a device-level auxiliary loss to encourage assigning equal computation load to all devices. [90] proposes a novel architecture that intrinsically achieves load balancing across devices. By partitioning experts into device-specific groups and enforcing a routing strategy that activates a fixed number of experts per group, the architectural design ensures balanced computational workloads.

Expert Capacity: Although auxiliary loss provides a gradient-based soft method for load balancing, the physical VRAM constraints necessitate enforced load balancing

mechanism. Therefore, expert capacity [24, 47] is introduced. Expert capacity refers to the threshold of how many tokens an expert can process. It is calculated by evenly distributing the number of tokens in a batch across the number of experts, and then multiplying by a capacity factor. A capacity factor greater than 1.0 creates additional buffer to accommodate for when tokens are not perfectly balanced across experts. If too many tokens are routed to an expert, the excess tokens are considered overflow, have their computation skipped, and are sent to the next layer via residual connections or completely discarded. [24] defines expert capacity as:

$$expert\ capacity = \left(\frac{T}{E}\right) f \tag{9}$$

Where T is the number of tokens, E is the total number of experts, and f is a free hyperparameter called the capacity factor. As shown in Figure 4, the capacity factor f introduces a trade-off: If the capacity factor is too large, this wastes computational resources through excessive padding. Conversely, if the capacity factor is too small, this sacrifices model performance due to token discards.

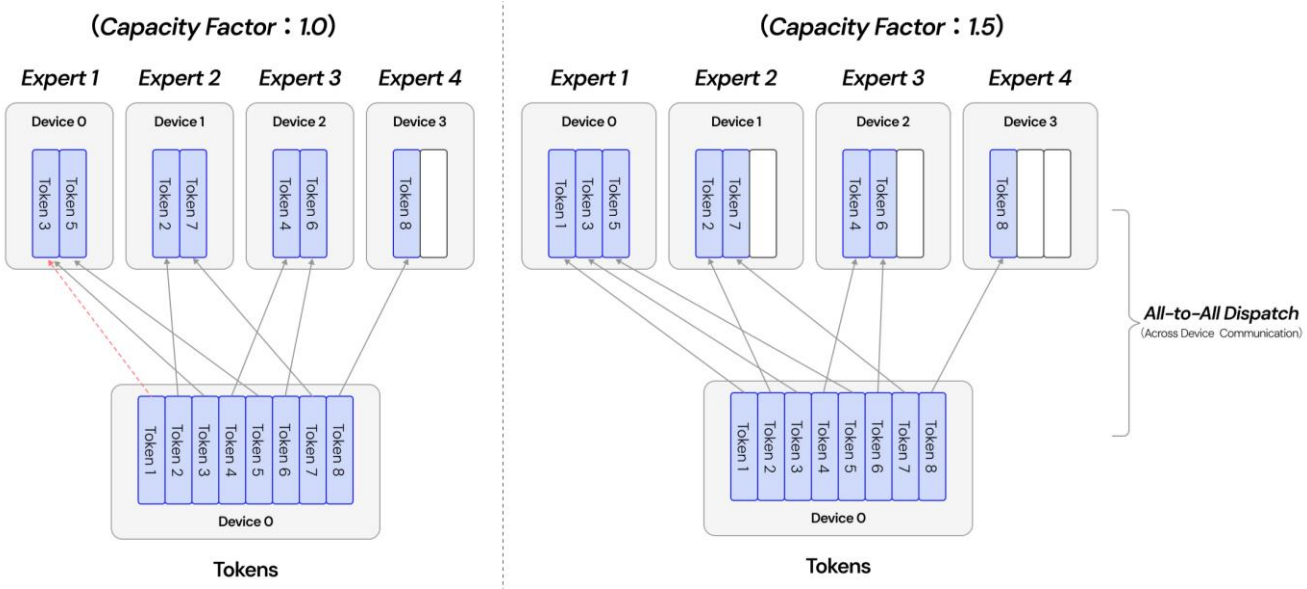

Figure 4. Illustration of token routing dynamics. The tokens are split across all devices using data parallelism. The experts are split across different devices, and each expert processes a fixed expert capacity of tokens modulated by the capacity factor. For a clear and intuitive view, the figure only shows the All-to-All Dispatch of Device 0. With $T = 8$, $E = 4$, it allows the router to send up to 2 tokens per expert. When the capacity factor f is 1.0, it must discard one token (red arrow). When the capacity factor f is 1.5, it needs to add excessive padding (white rectangles/empty slots).

2.2. Expert Network

The expert network is a core component of the MoE architecture. By splitting the dense FFN layers into multiple independent parts, each of which is an independent neural network—expert network [84]. Each expert possesses specialized knowledge in distinct domains and handles specific computational tasks and data subsets. These expert networks collectively accomplish a given task by each processing dedicated partitions of the input data. In practice, these experts are typically FFN units, but they can also be CNN [58, 102] or more complex network structures.

2.2.1. Shared Expert

Unlike conventional MoE, where all experts are designed independently, DeepSeekMoE [16] introduces a fine-grained experts method. The model further divides the expert network into shared experts and router experts. Shared experts are responsible for processing general features of all tokens, while routing experts dynamically allocate tokens based on their specific characteristics. This division not only reduces model redundancy and improves computational efficiency, but also ensures that shared experts capture common knowledge. The number of shared experts is fixed and always active, as shown in Figure 5.

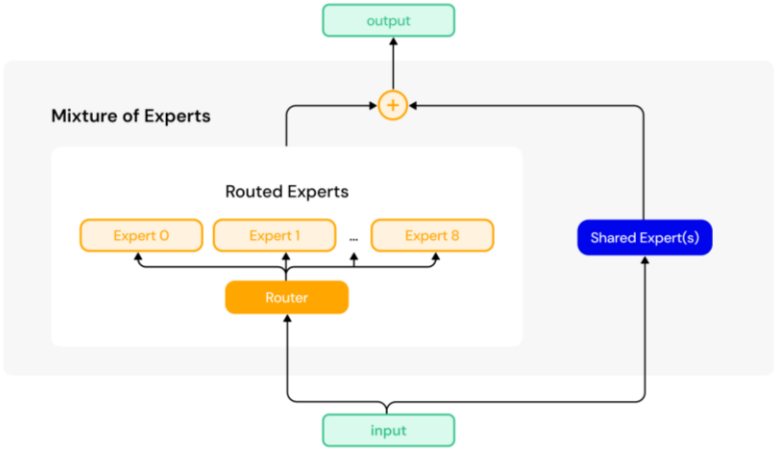

Figure 5. Expert network architecture with shared expert

In conventional routing strategies, tokens assigned to different experts may require shared knowledge or information. As a result, multiple experts might converge to learn the same shared knowledge in their parameters, leading to parameter redundancy. However, if there are dedicated experts responsible for capturing and sharing knowledge, this redundancy across different contexts can be alleviated. This reduction in redundancy helps build more parameter-efficient models [62, 75, 89, 92, 93].

2.2.2. Interpretability

MoE models more naturally lend themselves to interpretability studies [5, 66, 70], because each input is processed by an identifiable, discrete subset of the model weights. For example, some experts may specialize in handling punctuation marks, while others focus on proper nouns or similar elements. Additionally, researchers conducted multilingual training on the model [46, 129].

Interpretability of MoE models has not only been limited to text. [66] is a multi-modal model that was observed to learn experts that specialize in textual and visual data, including patches of textures, plants, eyes, and words.

3. Centralized and Decentralized Paradigm

As the remarkable capabilities of large models have elevated artificial intelligence technology to a new level, the AI industry has entered a new round of intense competition, which is becoming increasingly fierce. In the current AI field, the scale of training data and model parameters is showing a growing trend. The latest generation of MoE models that have larger numbers of parameters and longer context windows, which enables them to perform more complex cognitive tasks across a larger knowledge base.

Some of the most recent models [54, 92, 116], however, have scaled to 1 trillion parameters, have context windows that exceed 128K tokens, and have multiple feedforward networks that can operate independently. These models cannot fit on a single GPU, which means that the models must be chopped into smaller chunks and parallelized across multiple GPUs.

Traditional training and inference of MoE models are distributed on high-performance dedicated computing clusters via high-speed Remote Direct Memory Access (RDMA) network. Most of current advanced and high-efficient frameworks [45, 48, 77, 85, 127] focus on training and inference in homogeneous data center environments, are referred to as centralized paradigm. However, for resource-limited of individual researchers and small laboratories, even for well-resourced organizations, the prohibitive costs of high-performance dedicated computing clusters are unaffordable. Only a handful of large R&D corporations and institutions possess sufficient computational resources to develop advanced MoE models, leading to a quasi-monopoly that hindering people who lack large-scale high-end GPUs from training or deploying MoE models. In fact, consumer-level GPUs, which constitute a larger market share, are typically overlooked due to their weaker computing performance and lower communication bandwidth. Additionally, users may have privacy concerns. Decentralized paradigm has emerged as a promising paradigm to leverage dispersed resources across individual consumer-level GPUs and clusters, offering the potential to democratize MoE models development for broader communities. Collaborative computing allow for a much faster, smarter and scalable way of handling complex tasks. In this section, beyond centralized paradigm, we will delve into decentralized paradigm unlocking the potential collaborative computing of training and inference of MoE models with privacy protection in heterogeneous environments, while delineating critical challenges faced by the decentralized paradigm and reviewing existing related research efforts addressing these issues. Notably, efficient centralized paradigm methods remain essential in the decentralized paradigm, where resource constraints pose greater challenges.

3.1. Centralized Paradigm

Currently, MoE models training and inference primarily employ centralized resources deployed on high-performance dedicated computing clusters interconnected via high-speed RDMA networks. Nevertheless, efficient resource utilization remains challenging even in dedicated centralized infrastructures. In this section, we will comprehensively review the state-of-the-art methods for optimizing resource utilization. Table 3 presents an overview of highly efficient centralized frameworks that employ advanced resource optimization methods.

Table 3. An overview of the highly efficient centralized frameworks

Papers	Hybrid Parallelization				Time
	Expert Parallelism	Data Parallelism	Pipeline Parallelism	Tersor Parallelism	
Switch [24]	✓	✓		✓	2022.01
FasterMoE [32]	✓	✓	✓		2022.03
Megablocks [26]	✓			✓	2023.05
Tutel [36]	✓	✓		✓	2023.05
DeepSpeed-TED [86]	✓	✓		✓	2023.06
SmartMoE [123]	✓	✓	✓	✓	2023.07
vLLM [45]	✓	✓	✓	✓	2023.10
ColossalAI-MoE (OpenMoE) [115]	✓	✓		✓	2024.02
Skywork-MoE [103]	✓			✓	2024.06
DeepEP [127]	✓				2025.02

3.1.1. Expert Parallelism

Distributed training becomes a must to train MoE models, as the model is so large that it cannot be held in the memory of any single device. To support the distributed training,

GShard[47] designs a specific method of parallelism for MoE models, namely expert parallelism(EP). EP assigns different experts to distinct computing devices. Each device sends its own data to the device where the desired experts reside based on the MoE model's routing rules. This greatly reduces the number of parameters that each computation must interact with, as some experts are skipped. For non-MoE layers, EP behaves the same as common parallelism. As shown in Figure 6, the model is split up across the dimension of the experts' indices, and the input and output features are split along sample dimension. During EP, all-to-all communication is performed to dispatch the input samples to the desired expert models and put the output back to its original location, however, as the scale and frequency of all-to-all communication grow exponentially, the communication time increases significantly, resulting in reduced overall training efficiency. [127] develops a communication library tailored for MoE and EP to alleviate all-to-all communication bottlenecks, particularly among GPUs. [101] introduces Asynchronous Expert Parallelism (AEP), a new paradigm that decouples layer execution from barrier-style synchronization, effectively addresses the GPU underutilization and synchronization bottlenecks that commonly arise in expert parallel MoE serving.

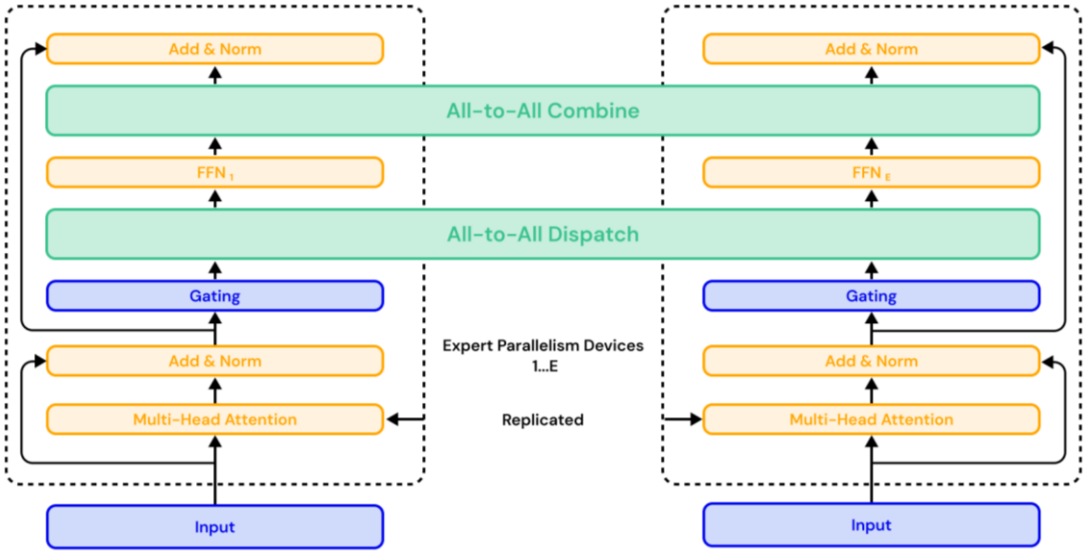

Figure 6. Expert parallelism for the MoE transformer block

3.1.2. Hybrid Parallelization

To further improve the performance of expert parallelism, Hybrid Parallelism has been extensively studied [24, 26, 32, 44, 86, 103, 115, 123], which combines a few of the below parallel strategies to better fit specific models and particular training hardware. Listed below are three common ways of parallelism.

Data Parallelism (DP): The data parallelism (DP) method [49, 78] hosts multiple copies of the MoE model on different GPUs or GPU clusters. Forward and backward computation are completed independently on each GPU. Gradients on different GPUs are aggregated before being used in the optimization of the model. However, DP alone is usually not sufficient with the latest generations of MoE modesl, as their model weights don't fit on a single GPU memory.

Pipeline Parallelism (PP): The model is divided into multiple stages and executes them sequentially on different devices [33, 67]. Each devices stores the parameters of its corresponding stage. The first device reads batches of the data, and devices with adjacent stages exchange intermediate results for forward or gradients for backward computation. To be efficient, PP has to improve the pipeline scheduling of devices to reduce idle waiting time, known as bubble time—a problem that has been intensively studied by prior works [8, 28, 30, 68, 74]. However, the layer-wise dependency of forward and backward processes limits the scalability of PP [29].

Tensor Parallelism (TP): It vertically splits model stages across multiple devices, each device stores a part of the parameters of the operators and conducts part of its computation, e.g. a tile of a matrix. TP of different operators needs to be designed specifically by experts, and the partitioning method is critical to distributed training performance. Megatron [85] provides the best practice of 1D TP on transformer models. However, 1D TP imposes higher demands on communication speed. Subsequent studies [9, 99, 112] have sought to refine this approach for most efficient, and these enhanced tensor parallelism algorithms demonstrate seamless compatibility with pipeline parallelism methods. For Transformer-based models, TP can enhance processing capabilities by allocating more GPU resources, speeding up processing.

3.2. Decentralized Paradigm

Decentralized paradigm leverages a broader range of resources compared to traditional centralized paradigm, yet it still employ mentioned above similar parallel strategies. The traditional centralized paradigm, a MoE system defines a machine learning model comprising multiple "experts", where each expert specializes in solving individual tasks. In other words, it splits complex models or tasks among smaller, more specialized networks. Each expert is trained on a specific aspect of the bigger task or data subset. A decentralized mixture of experts (DMoE) system takes it a step further. Instead of one central "boss" deciding which expert to use, multiple smaller systems each make their own decisions. This means the system can handle tasks more efficiently across different parts of a large system. The decentralized computing architecture enables the DMoE model to operate across laptops, workstations, and data centers by letting each part work independently, thus making it faster and more scalable, as depicted in Figure 7.

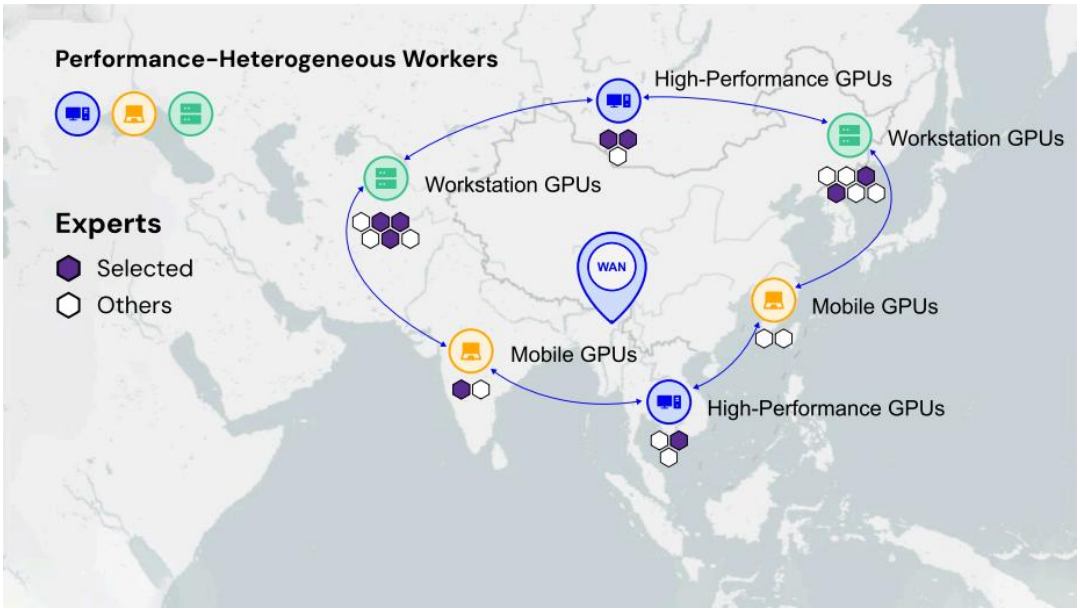

Figure 7. High-level scheme of decentralized MoE

However, compared with training and deploying MoE models in data centers equipped with high-ends GPUs, decentralized paradigm encounters several critical challenges, encompassing hardware performance heterogeneity, limited communication bandwidth, fault tolerance, security and privacy protection. In the subsequent discussion, we illustrate these challenges, concurrently reviewing existing related research efforts addressing these issues, as summarized in Table 4.

Table 4. An overview of critical challenges in the decentralized paradigm

Challenges	Works	Key features
Performance Heterogeneity	DMoE [82], DeDLOC [18], Petals [10], FusionAI [91], ATOM [106], Ravnest [61], HeterMoE [107]	Ranging from mobile GPUs to mid-end and high-performance GPUs.
Communication Efficiency	Saurabh Agarwal et al. [4], Moshpit [81], SWARM [80], Diloco [19], Opendiloc [38], Dilocox [73], StellaTrain [52]	Instead of high-speed RDMA network, compute nodes communicate mainly through LANs or even WANs with bandwidths under 10 Gb/s.
Fault Tolerance	Kademlia [59]	A compute node can join at any time, while existing nodes may fail to process a task for a variety of reasons.
Security and Privacy	Zhizhi Peng et al. [71], ZK-LLMs [105], PC-MoE [126], Hongyang Zhang et al. [124], Zhibo Xing et al. [111]	Vulnerable to attacks from malicious compute nodes that can disrupt system operations and compromise model robustness.

3.2.1. Hardware Performance Heterogeneity

In decentralized paradigms, when aggregating resources across consumer-grade GPUs, multiple nodes, or clusters, these computational devices are typically configured with various memory sizes, and communication bandwidths, ranging from mobile GPUs to mid-end and high-performance GPUs, these resources exhibit high heterogeneity. To prevent slower ones from becoming bottlenecks significantly, [82] proposes Decentralized Mixture of Experts (DMoE)-a layer designed for training with vast amounts of unreliable consumer-grade devices, different consumer-grade devices host varying numbers or scales of experts according to its available memory and compute capacity. [18] designs a novel algorithm framework named DeDLOC which combines parameter servers [6], All-Reduce SGD [83], decentralized SGD [51] and BytePS [42]. DeDLOC can accommodate a large number of heterogeneous devices with uneven compute capabilities for collaborative training. To address hardware heterogeneity challenges in decentralized training, Petals [10] proposes server dynamically select responsible layers based on available GPU memory. This is coupled with 8-bit quantization and load balancing to mitigate discrepancies in memory and computational resources on heterogeneous devices. [91] dissects directed acyclic graphs (DAGs) of model execution into sub-DAGs and loading them onto devices with limited memory. [106] presents ATOM, a resilient distributed training framework designed for asynchronous training of vast models in a decentralized setting using cost-effective hardware, including consumer-grade GPUs and Ethernet. The motivation behind ATOM's design is that a complete model can be executed on a single GPU layer by layer via memory swapping. By profiling individual model layers in detail, it devises an optimal model swapping schedule that effectively addresses the swapping overhead issue. [61] directly tackles bottlenecks arising from heterogeneous device performance during decentralized training. It proposes a "cluster-first, parallelize-later" strategy that employs genetic algorithm-based clustering to group heterogeneous nodes into capability-similar clusters according to RAM and bandwidth metrics, implements bubble-free asynchronous parallelism within clusters to eliminate straggler-induced idleness through asynchronous model execution, and facilitates efficient cross-cluster synchronization via multi-ring All-Reduce communication with parallelized ring coordination. [107] presents HeterMoE, a system to efficiently train MoE models on heterogeneous GPUs with no extra communication. HeterMoE disaggregates attention and expert modules to fully utilize each GPU's

capability. HeterMoE introduces zebra parallelism (ZP), along with asymmetric expert assignment (Asym-EA), to enable computation overlapping and fine-grained load balancing.

3.2.2. Communication Efficiency

RDMA technology has found extensive deployment in modern data centers, offering low latency and high throughput benefits that are particularly advantageous for generative AI training. Typically, centralized training is carried out in datacenter-grade GPU clusters equipped with homogeneous high-speed RDMA network, and network can reach over 400 Gb/s. In contrast, Decentralized training communicate through LANs or even WANs with bandwidths under 10 Gb/s. This will make training extremely slow, as gradient exchange is often bottlenecked by scarce network bandwidth, eventually leading to GPU underutilization [4]. Such bottlenecks are even worse in hybrid cluster settings when train a single model collaboratively across multiple geographically distributed consumer-grade GPUs, cloud GPU instances separated by the WANs with constrained and highly variable bandwidth.

Due to bandwidth limitations of LANs and WANs, it is essential to optimize communication efficiency for decentralized training to alleviate communication bottlenecks. [81] proposes a "Dynamic Grouping-AllReduce" mechanism that dynamically group active nodes into small-scale clusters. Within each cluster, it performs bandwidth-optimal AllReduce for gradient averaging, then iteratively propagates results across clusters to alleviate bandwidth constraints in decentralized environments. [80] proposes the SWARM parallel algorithm, the first decentralized model parallelism approach that leverages stochastic fault-tolerant pipelining and dynamic rebalancing to enhance communication efficiency. This approach deploys multiple candidate devices per pipeline-parallel stage. When a device outperforms others, it concurrently processes inputs from multiple slower predecessors and distributes outputs to multiple slower successors, maximizing bandwidth utilization. [19, 38] introduce two key innovations to integrate decentralized computing resources into a virtual supercomputer. One is low-frequency synchronization that global synchronization occurs only every H steps, maintaining parameter stability via momentum updates, and the other is decoupled inner/outer optimization that local replicas independently perform H-step internal optimization (e.g., AdamW), then aggregate updates through an external optimizer (e.g., Nesterov momentum SGD). This simultaneously reduces communication requency and enhances tolerance to large batch sizes. [73] enables low-communication decentralized cluster training for up to 100-billion-parameter models by integrating gradient compression with overlapped communication and local training.

presents StellaTrain, the first framework for distributed training that minimizes the communication intensity of model training in multi-cluster environments separated by a WAN. StellaTrain introduces two key enablers to achieve such high training speeds. First, StellaTrain employs gradient compression to effectively use the network in low-bandwidth environments and exploits the resulting sparsity of gradients to devise computationally efficient compression and optimization. Second, a layer-wise partial staleness mechanism is designed in StellaTrain, where some layers receive gradient updates immediately, while others are delayed by one iteration.

3.2.3. Fault Tolerance

With distributed decision-making, the system can continue functioning even if one gate or expert fails. This resilience ensures uninterrupted operation and minimizes the risk of complete system failure. Given the inherent instability of computational resources contributed by communities, node can join at any time, while existing nodes can exit due to various reasons. Moreover, node failures and communication disruptions are inevitable in decentralized infrastructure. Consequently, fault tolerance becomes a critical requirement to ensure stable and efficient training processes.

To implement fault tolerance mechanisms in a decentralized system, distributed hash table (DHT) have emerged as a core technology. This is a family of distributed data structures that store key-value pairs across multiple computers in a network. A single computer within such structure only needs to "know" O (log N) out of N computers, at the same time it can look up any key with at most O(log N) requests to its peers. There are several DHT variants, but they all have common properties:

Decentralization: Nodes form and maintain DHT without any central coordination.

Scalability: DHT can scale to millions of active nodes that are continually joining and leaving.

Fault tolerance: A failure in one or a few nodes does not affect DHT integrity and availability.

By far, the most popular DHT variation is Kademlia [59] with numerous applications such as BitTorrent, I2P, and Ethereum.

3.2.4. Security and Privacy Protection

With the rapid advancement of generative AI technologies, data privacy and model security in decentralized environment have become critical challenges. The potential risks of infringement privacy are the unauthorized data access and manipulation. There is also the threat of malicious computing nodes that can disrupt system operations and compromise model robustness. The issue of membership inference can expose if a specific data point was used in training the model. In response to these challenges, privacy-preserving techniques such as secure multiparty computation, differential privacy, and encryption are suggested as mitigation strategies.

Zero-Knowledge Machine Learning (ZKML) is an emerging technology that applies Zero-Knowledge Proofs (ZKP) to the machine learning domain [71]. It enables data owners to utilize their data for training machine learning models without sharing raw data with third parties. This approach ensures data privacy and mitigates the risk of data breaches. Simultaneously, it allows data owners to selectively share model outcomes, thereby reconciling data security requirements with machine learning operational needs. It offers a viable solution for enhancing security and privacy protection in decentralized environment [105, 111, 124].

Proposes Privacy-preserving Collaborative Mixture-of-Experts (PC-MoE) [126], which enables multiple parties to leverage each other's compute resources and data to reduce the hardware burden and improve their own MoE model's performance without sacrificing data privacy. It demonstrates that preserving privacy does not always have to come at a steep cost to utility or vice versa: by routing only sparse expert signals, parties obtain near-centralized performance and enjoy lower hardware requirements, while revealing virtually nothing to an attacker.

These findings motivate future privacy research to explore

how far the utility frontier can be pushed with minimal compromise in privacy and safety.

4. DownStream Applications

In recent years, MoE technology has witnessed exponential advancement, with deployments now spanning NLP [62, 94], CV [79, 125], multimodal systems [31, 50, 53, 66, 108], and intelligent agent frameworks, leveraging its ability to dynamically allocate computational resources and specialize in different data distributions, demonstrating revolutionary efficiency advantages [54]. In the following, we will focus on exploring the typical vertical domains applications of MoE, including medical-assisted diagnosis, autonomous driving, financial analysis, business intelligence, and blockchain, as illustrated in Figure 8. The aim is to provide an overall understanding of how MoE can be utilized for specific tasks. These application scenarios are also more appropriate for decentralized infrastructures, where resource constraints are more severe and complex.

Figure 8. An overview of MoE applications in typical vertical domains

4.1. Medical Diagnosis

Recent advancements in general-purpose or domain-specific multimodal models have witnessed remarkable progress for medical decision-making. However, medical data is inherently heterogeneous across different resolutions, modalities and clinical centers [34], posing unique challenges for developing generalizable foundation models. Conventional entails training distinct models per dataset or using a shared encoder with modality-specific decoders, but these approaches incur heavy computational overheads and suffer from poor scalability, hindering their clinical utility across diverse resource-constrained scenarios in practice.

Proposes a lightweight framework for multimodal medical tasks [40], addressing both discriminative and generative needs. Optimized for resource-constrained environments, it involves aligning medical images with language model tokens, task-specific instruction tuning, and domain-specific expert fine-tuning. From coarse global patterns to fine-grained localized structures processing framework is introduced in [15], a vision–language model that dynamically routes multi-scale visual features through a diagnosis-conditioned MoE framework. This framework produces localized visual representations aligned with textual descriptions, without requiring modality-specific supervision at inference, improves the performance of modality-specialized visual representations in clinical vision-language systems. [56] introduces a novel approach to improve medical text embeddings tasks for Retrieval-Augmented Generation (RAG) in specialized medicine domains. Med-MoE-Embed leverages a pretrained embedding backbone augmented with a trainable MoE network, allowing for efficient adaptation to specific medical subdomains and tasks, significant improvements in retrieval accuracy and generation quality, enhancing the model's adaptability across various medical datasets. Med-MoE-Embed mitigates the challenges of limited data accessibility and domain-specific requirements in the medical field, offering a versatile and efficient solution for enhancing embedding quality in medical natural language processing applications. [110] proposes a decentralized federated learning framework named dFLMoE for medical data analysis. In this framework, clients directly exchange lightweight head models with each other. After exchanging, each client treats both local and received head models as individual experts, and utilizes a client-specific MoE approach to make collective decisions. This design not only protects patients' privacy but also removes the dependency on the central server to enhance the robustness of the framework.

Introduces MoE and specifically couple it with a sparse gating network model to handle patient heterogeneous electronic health record (EHR) for risk prediction.it enhance the effectiveness of risk prediction [34]. [121] addresses the difficulties of MoE models deploy on the Internet of Medical Things (IoMT) for individuals' personalized e-healthcare. By designing a new medical LLM based on the MoE architecture with an offloading strategy, meeting the deployment resource demands in the IoMT, improving e-healthcare services and the privacy protection for users.

Focus on medical image segmentation base on MoE. Notably [13, 41, 57], [41] proposes the Medical Multimodal

Mixture of Experts (M4oE) framework named M4oE, leveraging the SwinUNet architecture. Specifically, M4oE comprises modality-specific experts, each separately initialized to learn features encoding domain knowledge. Subsequently, a gating network is integrated during fine-tuning to modulate each expert's contribution to the collective predictions dynamically. This enhances model interpretability and generalization ability while retaining expertise specialization. Simultaneously, the M4oE architecture amplifies the model's parallel processing capabilities, and it also ensures the model's adaptation to new modalities with ease. Moreover, M4oE showcases a significant reduction in training overhead.

4.2. Autonomous Driving

The great success of sparsely-gated MoE in various fields has inspired their application in autonomous driving. End-to-end learning from sensory data has shown promising results in autonomous driving. While employing many sensors enhances world perception and should lead to more robust and reliable behavior of autonomous vehicles, it is challenging to train and deploy such network and at least two problems are encountered in the considered setting. The first one is the heterogeneous data typically originates from various sensors, systems, exhibiting a high degree of heterogeneity and diversity. The other is the increase of computational complexity with the number of sensing devices.

Proposes a Multi-modal Experts Network [22], where each sensor is paired with a lightweight expert sub-network, and introduce a two-level gating network to handle inputs coming from three cameras and one LiDAR. The gating network chooses the camera input in a discrete way from among several mutually-exclusive sensors. Alternatively, the network chooses the LiDAR sensor, which covers the same field of view as the camera sensors, and identifies continuously in real-time the part of its depth map with a narrow field of view that is useful for steering autonomously. On resource-constrained in-vehicle platforms, the proposed multi-modal experts network to perform conditional computation that converts multi-sensor redundancy into robustness rather than a computational burden, thereby addressing the fundamental challenges of multi-sensor fusion in autonomous driving systems. [64] presents MIxture of eXperts Odometry (MIXO), a data-driven, machine learning-based technique that loosely combines odometry outputs from multiple cameras to obtain a more accurate and robust global estimate. In MIXO, each camera (expert) is individually processed by a state-of-the-art visual odometry algorithm. Then, the odometry estimates are mixed by a gating network, which selects the locally optimal experts in the current operational conditions and weights their contributions accordingly. MIXO achieves more robust and accurate results than any single camera, reducing the absolute rotational and translation error. [113] introduces a framework that integrates the MoE paradigm into LiDAR data representation learning to synergistically combine multiple representations, such as range images, sparse voxels, and raw points, captures complementary information from different representations, enabling more robust scene understanding.

Is the first work to successfully apply the MoE mechanism to joint prediction-and-planning modeling for safe autonomous driving on public roads [72]. By learning and selecting from a distribution of expert trajectories, it captures multimodal future trajectory distributions, thereby enhancing system safety and robustness in complex traffic scenarios. Compared with traditional approaches, MoE delivers several advantages: strong multimodal modeling capabilities that adapt to intricate traffic situations, a data-driven safety strategy that obviates rule-based systems and is readily scalable, and high interpretability. [25] introduces autoregressive end-to-end trajectory planning with MoE for autonomous driving, it addresses traditional static planners' inability to model temporal dependencies, named Artemis. Through its integrated MoE architecture with dedicated routing networks, Artemis dynamically captures the intrinsic dynamic characteristics of driving behavior and effectively accommodates diverse driving environments. The framework further incorporates a lightweight batch reallocation strategy, significantly enhancing both training efficiency and deployment viability. [87] applies the MoE architecture to end-to-end motion planning tasks. Through an expert routing mechanism, it dynamically selects specialized expert sub-networks to resolve conflicts and trade-offs between different driving objectives (e.g., obstacle avoidance, yielding, lane keeping), significantly improves generalization across diverse driving scenarios, including challenging out-of-distribution zero-shot cases. This is the first introduction of expert specialization for learning driving reward preferences, effectively resolving "modality collapse" and "reward balancing" conflicts. The resulting scalable motion planning model scales up to 800M parameters, employs 8 experts with top-2 expert activation, and supports seamless scaling from small (100M) to large (1B) models following scaling law. [117] introduces both a Scene-Specialized Vision MoE and a Skill-Specialized Action MoE, specifically designed for end-to-end autonomous driving scenarios, named DriveMoE. DriveMoE dynamically selects contextually relevant camera views and activates skill-specific experts for specialized planning. The Vision MoE employs a learned router to dynamically prioritize camera views aligned with the immediate driving context, integrating projector layers that fuse these selected views into a cohesive visual representation. Concurrently, the Action MoE leverages another routing mechanism to engage distinct experts within a flow-matching planning architecture, with each expert dedicated to handling specialized behaviors such as lane following, obstacle avoidance, or aggressive maneuvers. By introducing context-driven dynamic expert selection across both perception and planning modules, DriveMoE significantly enhancing computational efficiency and robustness to rare.

Designs an innovative multi-task Driver Monitoring System (DMS) [100], termed VDMoE, which leverages expert specialization and task-level gating networks, the model achieves state-of-the-art performance on drowsiness, cognitive load, heart rate, and respiration rate estimation, simultaneously delivering high accuracy, efficiency, and interpretability, thereby providing a reliable monitoring solution for autonomous driving.

4.3. Financial and Business Analysis

MoE have revolutionized data analysis domains due to its ability to handle heterogeneous data sources and dynamic changes [98]. For intricate financial and business analysis tasks, MoE tailored to specific domains can achieve a more comprehensive understanding and more superior insights by routing different factors to specialized expert networks and the gating network dynamically coordinates the outputs of the experts. With the scale and complexity of financial and

business data have increased significantly, and traditional analysis tools are struggling to cope with this. Financial and business institutions have gradually adopted the MoE architecture to build an automated data science workflow for improve the accuracy and efficiency of data analysis tasks, helping them effectively process market movement forecasting, risk assessment making, portfolio optimization, fraud detection, customer preference mining, etc. For example, the quantitative investment field is highly dependent on accurate stock forecasting and profitable investment decision-making [88]. This role-specific specialization enhances the model's ability to integrate their domain-specific expertise.

Introduced FinTeamExperts [119], a novel framework of role-specialized LLM designed as a MOE to excel in financial analysis tasks. By mimicking a real-world team setting within the finance domain, each model in FinTeamExperts specializes in one of three critical roles: Macro Analysts, Portfolio Managers, and Quantitative Analysts. The results showcase the potential of advanced LLM in transforming financial analysis and decision-making, forming a comprehensive and robust financial analysis tool, paving the way for more sophisticated and practical AI applications in the finance industry.

Offers an extensible MoE framework for financial time-series modeling [95]. It applies MoE to stock-price forecasting, combines an RNN for volatile stocks and a linear model for stable stocks, dynamically adjusting the weight of each model through a gating network, significantly improves predictive accuracy across different volatility profiles. [109] designs a Multi-field-aware Mixture-of-Experts (MfMoE) architecture which can simultaneously learn the single-field and global-field information, significantly improve the accuracy of default prediction in the financial field, effectively solving the problems of numerical feature encoding and high-order feature interaction modeling.

Introduces a Mixture-of-Experts-based deep reinforcement learning portfolio model (MoEDRLPM) that addresses limitations in spatio temporal modeling and strategy diversity in traditional investment approaches by dynamically selecting the current optimal expert from the mixed expert pool through router [104]. The expert makes decisions and aggregates to derive the portfolio weights. Notably, the model achieves significant improvements in return metrics. These results robustly demonstrate MoE's practical value in financial decision-making scenarios—where expert specialization simultaneously optimizes returns and risk control.

Understanding consumer choice is fundamental to marketing and management research, since it allows a business to make better use of marketing budgets as well as to gain a competitive edge over rival companies. More importantly, it demonstrates better knowledge of various customer purchasing behaviors and patterns over time. Some of the challenges faced by e-commerce, stores, and supermarkets involve dealing with huge volumes of customers with different and similar wants, different and similar purchase prices, and buying patterns. [7, 96] focus on customer mining technique based on MoE, as a machine learning-driven alternative that dynamically segments consumers based on latent behavioral patterns. By leveraging probabilistic gating functions and specialized expert networks, MoE provides a flexible, nonparametric approach to modeling heterogeneous preferences, significantly enhances predictive accuracy over traditional econometric models, capturing nonlinear consumer responses to price variations, brand preferences, and product attributes. The findings underscore MoEs potential to improve demand forecasting, optimize targeted marketing strategies, and refine segmentation practices. These studies bridges the gap between data-driven MoE approaches and marketing theory, advocating for the integration of AI techniques in managerial decision-making and strategic consumer insights. As markets continue to evolve and consumer preferences become increasingly complex and dynamic, the MoE framework provides a powerful tool for understanding and predicting economic behavior in an era of data-driven decision-making.

4.4. Blockchain

A DMoE model adapts MoE to a decentralized network, such as a blockchain [14]. This means that rather than a central entity controlling the experts, the decision-making and control are spread across multiple smaller systems which hosted on peer devices. Simply put, the network autonomously selects the most suitable expert (a node or smart contract) based on what the task needs.

Although the MoE model has a long history and achieved great success in many fields during the past few years [21, 23, 37, 39, 122], the intersection of MoE and blockchain is largely under-explored, MoE can still play a role in several aspects of blockchain technology. Blockchain is a innovative decentralized, distributed ledger technology that enables secure and transparent transactions without the need for intermediaries. By utilizing advanced cryptographic methods, blockchain ensures the integrity and verification of each transaction, establishing a highly reliable technological framework. Within this ecosystem, smart contracts function as self-executing programs on the blockchain, automating the management of digital assets such as cryptocurrencies. These contracts are activated when specific conditions are met and, once deployed, become permanent components of the blockchain.

Proposes a smart contract vulnerability detection framework based on mixture-of-experts tuning (MOE-Tuning) of LLM named MOS [120]. Through the MoE architecture establishes a "Dynamic Routing-Specialized Experts" collaborative framework that reconstructs the paradigm for smart contract security detection, significantly outperforms existing state-of-the-art methods with improvements in accuracy. The vulnerability explanations generated by MOS also demonstrate high quality, achieving positive ratings for correctness, completeness, and conciseness, respectively, through a combined approach of human evaluation and LLM evaluation. It demonstrates the effectiveness and scalability of MoE in smart contract security detection. SAEL [118] further introduce an Adaptive Mixture-of-Experts architecture mechanism which dynamically adjusts feature weights via a Gating Network to enhance the performance of smart contract vulnerability detection.

5. Challenges and Opportunities

Although sparse MoE models have undergone continuous exploration and innovation, achieving significant advancements, the main technical challenges still remain. The following are some of the key challenges and promising research directions we have considered.

Load balancing: Load balancing is a critical issue in MoE models training. However, load balancing in MoE is a double-edged sword. Overly pursuing balance can hinder the model's

expert specialization and expressive capacity, while neglecting it can lead to issues of training stability and resource wastage. Although current efforts have attempted to address this challenge by incorporating Expert-Level and Device-Level auxiliary loss to encourage tokens to be evenly distributed among experts across multiple devices, these strategies can still lead to training instability and often neglect the influence of model performance. Therefore, future work should focus on developing more effective strategies that ensure balanced utilization of experts and model training stability. Techniques such as adaptive load balancing, dynamic expert capacity adjustment, regularization methods and innovative gating algorithms could be explored.

Decentralized: Decentralized MoE is an exciting but their potential in combination with other technology paradigms remains underexplored, particularly when combining the principles of decentralization in blockchain with specialized AI models. While this combination holds potential, it also introduces a set of unique challenges that need to be addressed:

High Scalability: Distributing computational tasks across decentralized nodes can have highly heterogeneous GPUs, CPUs, and network bandwidth, ranging from mobile GPUs to mid-range GPUs in PCs or high-performance GPUs. Network performance, CPUs, storage, and other resources also exhibit high heterogeneity which will create load imbalances and network bottlenecks, limiting scalability. Efficient resource allocation is critical to support a large number of computing nodes to participate, resulting in a high acceleration ratio of system performance.

Fault Tolerance: Decentralized nodes can dynamically join and leave the system, which may cause failures in executing specific AI tasks. This requires that a failure in one or a few nodes does not affect system integrity and availability. Howerver, managing the variability of collaboration and synchronization of updates across distributed experts can lead to issues with model quality and fault tolerance.

Communication Latency: Decentralized nodes often connect via the internet, where communication bandwidth is significantly lower than in high-performance computing clusters. This results in substantial communication latency, especially when exchanging large volumes of data between nodes. Therefore, decentralized MoE systems may experience higher latency due to the need for inter-node communication, which may hinder real-time decision-making applications.

Security and Privacy Protection: Decentralized systems are more vulnerable to attacks. Protecting data privacy and ensuring expert integrity without a central control point is challenging. However, existing methods incur excessive computational overhead.

Incentive Mechanism: Incentive mechanism serve as economic catalysts to encourage users to participate in decentralized system, where nodes with larger contributions are rewarded more, preventing malicious nodes from gaining incentives. Compared with previous incentive schemes in other similar scenarios, there are some special challenges and considerations in the mechanisms design. The property of online training indicates that the arrival and departure time is unknown and varies drastically for different node. This online property is in line with asynchronous training but renders previous one-round incentive schemes ineffective. The incentive mechanism should be robust and resilient to some malicious clients which contribute nothing but endeavor to get large paybacks.

These challenges require innovative solutions in decentralized AI architectures, consensus algorithms and privacy-preserving techniques. Advances in these areas will be key to making decentralized MoE systems more scalable, efficient, democratization and secure, ensuring they can handle increasingly complex tasks in a distributed environment.

Expert Specialization: Expert specialization refers to a capability where each expert possesses non-overlapping and highly focused knowledge. Research has demonstrated that encouraging experts to concentrate their skills on specific subtasks or domains significantly enhances the performance and generalization ability of MoE models. However, researchers have found that the so-called "experts" in MoE tend to focus on specific types of tokens or shallow-level concepts. For example, some experts may specialize in handling punctuation marks, while others focus on proper nouns or similar elements. At most, they are grammar-level experts. In addition, the researchers conducted multilingual training on the model. Although one might expect each expert to handle a specific language domain, in fact, this is not the case. More specifically, their expertise lies in processing specific tokens within particular contexts, tending to focus on grammar rather than specific domains. The content that experts learn is more detailed than the broader specific domain. Therefore, future studies should focus on investigating novel mechanisms for enhancing the expert specialization for the development of more powerful MoE models. For instance, introducing meta-learning mechanism into the expert networks to enhance their adaptability and improve the model's ability to handle more complex tasks as well as the specialization of the experts.

6. Conclusion

In this survey, we introduce the theoretical foundations and core design elements of MoE, such as the sparse activation mechanism of expert networks, routing mechanism, load balancing, and extend beyond centralized paradigm to delve into decentralized paradigm. we then exploring its vertical domain applications including medical diagnosis, autonomous driving, financial analysis, business analysis, blockchain, and identify critical challenges and promising directions for future investigation. We hope that this survey serves as a valuable reference for researchers and practitioners, tracking the latest research developments and inspiring new ideas in this explosively evolving field.

References

[1] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, Et Al., Phi-3 technical report: A highly capable language model locally on your phone, 2024, arXiv preprint arXiv:2404.14219, (2024).

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, Et Al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774, (2023).

[3] S. Agarwal, L. Ahmad, J. Ai, S. Altman, A. Applebaum, E. Arbus, R. K. Arora, Y. Bai, B. Baker, H. Bao, Et Al., gpt-oss-120b & gpt-oss-20b model card, arXiv preprint arXiv:2508.10925, (2025).

[4] S. Agarwal, H. Wang, S. Venkataraman, And D. Papailiopoulos, On the utility of gradient compression in distributed training systems, Proceedings of Machine Learning and Systems, 2022, pp. 652–672.

[5] R. Akrour, D. Tateo, And J. Peters, Continuous action reinforcement learning from a mixture of interpretable experts, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, pp. 6795–6806.

[6] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, And B.-Y. Su, Scaling distributed machine learning with the parameter server, in Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation, USENIX Association, 2014, pp. 583–598.

[7] V. I. Anireh, E. N. Osegi, And A. Silas, A model for customer opinion mining and sentiment classification using a mixture of experts machine learning model, Computer Science, 2024, pp. 51–61.

[8] D. Arfeen, Z. Zhang, X. Fu, G. R. Ganger, And Y. Wang, Pipefill: Using gpus during bubbles in pipeline-parallel llm training, arXiv preprint arXiv:2410.07192, (2024).

[9] Z. Bian, Q. Xu, B. Wang, And Y. You, Maximizing parallelism in distributed training for huge neural networks. corr abs/2105.14450 (2021), arXiv preprint arXiv:2105.14450, (2021).

[10] A. Borzunov, D. Baranchuk, T. Dettmers, M. Ryabinin, Y. Belkada, A. Chumachenko, P. Samygin, And C. Raffel, Petals: Collaborative inference and fine-tuning of large models, arXiv preprint arXiv:2209.01188, (2022).

[11] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, Et Al., Language models are few-shot learners, Advances in neural information processing systems, 2020, pp. 1877–1901.

[12] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, And J. Huang, A survey on mixture of experts in large language models, IEEE Transactions on Knowledge and Data Engineering, 2025, pp. 3896-3915.

[13] Q. Chen, L. Zhu, H. He, X. Zhang, S. Zeng, Q. Ren, And Y. Lu, Low-rank mixture-of-experts for continual medical image segmentation, in International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2024, pp. 382–392.

[14] Z.-K. Chong, H. Ohsaki, And B. Ng, Llm-net: Democratizing llms-as-a-service through blockchain-based expert networks, arXiv preprint arXiv:2501.07288, (2025).

[15] S. Chopra, L. Mao, G. Sanchez-Rodriguez, A. J. Feola, J. Li, And Z. Kira, Medmoe: Modality-specialized mixture of experts for medical vision-language understanding, arXiv preprint arXiv:2506.08356, (2025).

[16] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, Et Al., Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, arXiv preprint arXiv:2401.06066, (2024).

[17] J. Dean, Introducing pathways: A next-generation ai architecture, 2021, URL https://blog.google/technology/ai/introducing-pathways-next-generation-ai-architecture.

[18] M. Diskin, A. Bukhtiyarov, M. Ryabinin, L. Saulnier, A. Sinitsin, D. Popov, D. V. Pyrkin, M. Kashirin, A. Borzunov, A. Villanova Del Moral, Et Al., Distributed deep learning in open collaborations, Advances in Neural Information Processing Systems, 2021, pp. 7879–7897.

[19] A. Douillard, Q. Feng, A. A. Rusu, R. Chhaparia, Y. Donchev, A. Kuncoro, M. Ranzato, A. Szlam, And J. Shen, Diloco: Distributed low-communication training of language models, arXiv preprint arXiv:2311.08105, (2023).

[20] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat, Et Al., Glam: Efficient scaling of language models with mixture-of-experts, in International conference on machine learning, PMLR, 2022, pp. 5547–5569.

[21] D. Eigen, M. Ranzato, And I. Sutskever, Learning factored representations in a deep mixture of experts, arXiv preprint arXiv:1312.4314, (2013).

[22] S. Fang And A. Choromanska, Multi-modal experts network for autonomous driving, in 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 6439–6445.

[23] W. Fedus, J. Dean, And B. Zoph, A review of sparse expert models in deep learning, arXiv preprint arXiv:2209.01667, (2022).

[24] W. Fedus, B. Zoph, And N. Shazeer, Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, Journal of Machine Learning Research, 2022, pp. 1–39.

[25] R. Feng, N. Xi, D. Chu, R. Wang, Z. Deng, A. Wang, L. Lu, J. Wang, And Y. Huang, Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving, arXiv preprint arXiv:2504.19580, (2025).

[26] T. Gale, D. Narayanan, C. Young, And M. Zaharia, Megablocks: Efficient sparse training with mixture-of-experts, Proceedings of Machine Learning and Systems, 2023, pp. 288–304.

[27] W. Gan, Z. Ning, Z. Qi, And P. S. Yu, Mixture of experts (moe): A big data perspective, arXiv preprint arXiv:2501.16352, (2025).

[28] L. Guan, D. Li, Y. Chen, J. Liang, W. Wang, And X. Lu, Pipeoptim: Ensuring effective 1f1b schedule with optimizer-dependent weight prediction, IEEE Transactions on Knowledge and Data Engineering, 2025, pp. 2831-2845.

[29] L. Guan, D.-S. Li, J.-Y. Liang, W.-J. Wang, K.-S. Ge, And X.-C. Lu, Advances of pipeline model parallelism for deep learning training: an overview, Journal of Computer Science and Technology, 2024, pp. 567–584.

[30] L. Guan, W. Yin, D. Li, And X. Lu, Xpipe: Efficient pipeline model parallelism for multi-gpu dnn training, arXiv preprint arXiv:1911.04610, (2019).

[31] X. Han, L. Wei, Z. Dou, Z. Wang, C. Qiang, X. He, Y. Sun, Z. Han, And Q. Tian, Vimoe: An empirical study of designing vision mixture-of-experts, arXiv preprint arXiv:2410.15732, (2024).

[32] J. He, J. Zhai, T. Antunes, H. Wang, F. Luo, S. Shi, And Q. Li, Fastermoe: modeling and optimizing training of large-scale dynamic pre-trained models, in Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, 2022, pp. 120–134.

[33] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, Et Al., Gpipe: Efficient training of giant neural networks using pipeline parallelism, Advances in neural information processing systems, 2019, pp. 103 - 112.

[34] Z. Huo, L. Zhang, R. Khera, S. Huang, X. Qian, Z. Wang, And B. J. Mortazavi, Sparse gated mixture-of-experts to separate and interpret patient heterogeneity in ehr data, in 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), IEEE, 2021, pp. 1–4.

[35] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, Et Al., Gpt-4o system card, arXiv preprint arXiv:2410.21276, (2024).

[36] C. Hwang, W. Cui, Y. Xiong, Z. Yang, Z. Liu, H. Hu, Z. Wang, R. Salas, J. Jose, P. Ram, Et Al., Tutel: Adaptive mixture-of-experts at scale, Proceedings of Machine Learning and Systems, 2023, pp. 269–287.

[37] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, And G. E. Hinton, Adaptive mixtures of local experts, Neural computation, 1991, pp. 79–87.

[38] S. Jaghouar, J. M. Ong, And J. Hagemann, Opendiloco: An open-source framework for globally distributed low-communication training, arXiv preprint arXiv:2407.07852, (2024).

[39] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. D. L. Casas, E. B. Hanna, F. Bressand, Et Al., Mixtral of experts, arXiv preprint arXiv:2401.04088, (2024).

[40] S. Jiang, T. Zheng, Y. Zhang, Y. Jin, L. Yuan, And Z. Liu, Med-moe: Mixture of domain-specific experts for lightweight medical vision-language models, arXiv preprint arXiv:2404.10237, (2024).

[41] Y. Jiang And Y. Shen, M4oe: A foundation model for medical multimodal image segmentation with mixture of experts, in International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2024, pp. 621–631.

[42] Y. Jiang, Y. Zhu, C. Lan, B. Yi, Y. Cui, And C. Guo, A unified architecture for accelerating distributed {DNN} training in heterogeneous {GPU/CPU} clusters, in 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20), 2020, pp. 463–479.

[43] J. Kaplan, S. Mccandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, And D. Amodei, Scaling laws for neural language models, arXiv preprint arXiv:2001.08361, (2020).

[44] Y. J. Kim, A. A. Awan, A. Muzio, A. F. C. Salinas, L. Lu, A. Hendy, S. Rajbhandari, Y. He, And H. H. Awadalla, Scalable and efficient moe training for multitask multilingual models, arXiv preprint arXiv:2109.10465, (2021).

[45] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, And I. Stoica, Efficient memory management for large language model serving with pagedattention, SOSP '23, New York, NY, USA, 2023, Association for Computing Machinery, pp. 611–626.

[46] V. N. Kyle Kranen, Applying mixture of experts in llm architectures, URL https://developer.nvidia.com/blog/applying-mixture-of-experts-in-llm-architectures, (2024).

[47] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, And Z. Chen, Gshard: Scaling giant models with conditional computation and automatic sharding, arXiv preprint arXiv:2006.16668, (2020).

[48] S. Li, H. Liu, Z. Bian, J. Fang, H. Huang, Y. Liu, B. Wang, And Y. You, Colossal-ai: A unified deep learning system for large-scale parallel training, in Proceedings of the 52nd International Conference on Parallel Processing, 2023, pp. 766–775.

[49] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, Et Al., Pytorch distributed: Experiences on accelerating data parallel training, arXiv preprint arXiv:2006.15704, (2020).

[50] Y. Li, S. Jiang, B. Hu, L. Wang, W. Zhong, W. Luo, L. Ma, And M. Zhang, Uni-moe: Scaling unified multimodal llms with mixture of experts, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2025, pp. 3424–3439.

[51] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, And J. Liu, Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent, Advances in neural information processing systems, 2017, pp. 5336 - 5346.

[52] H. Lim, J. Ye, S. Abdu Jyothi, And D. Han, Accelerating model training in multi-cluster environments with consumer-grade gpus, in Proceedings of the ACM SIGCOMM 2024 Conference, 2024, pp. 707–720.

[53] B. Lin, Z. Tang, Y. Ye, J. Cui, B. Zhu, P. Jin, J. Huang, J. Zhang, Y. Pang, M. Ning, Et Al., Moe-llava: Mixture of experts for large vision-language models, arXiv preprint arXiv:2401.15947, (2024).

[54] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, Et Al., Deepseek-v3 technical report, arXiv preprint arXiv:2412.19437, (2024).

[55] J. Liu, P. Tang, W. Wang, Y. Ren, X. Hou, P.-A. Heng, M. Guo, And C. Li, A survey on inference optimization techniques for mixture of experts models, arXiv preprint arXiv:2412.14219, (2024).

[56] X. Liu, Y. Zhu, T. Pang, K. Xue, X. Zhang, And C. Fan, Medical document embedding enhancement with heterogeneous mixture-of-experts, in 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2024, pp. 2238–2243.

[57] Y. Liu, J. Pei, Z. He, G. Yang, Z. Jiang, And Q. Lao, Medical language mixture of experts for improving medical image segmentation, in 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2024, pp. 2210–2216.

[58] Y. Lou, F. Xue, Z. Zheng, And Y. You, Sparse-mlp: A fully-mlp architecture with conditional computation, arXiv preprint arXiv:2109.02008, (2021).

[59] P. Maymounkov And D. Mazieres, Kademlia: A peer-to-peer information system based on the xor metric, in International workshop on peer-to-peer systems, Springer, 2002, pp. 53–65.

[60] D. Mcallister, M. Tancik, J. Song, And A. Kanazawa, Decentralized diffusion models, arXiv preprint arXiv:2501.05450, (2025).

[61] A. R. Menon, U. Menon, And K. Ahirwar, Ravnest: Decentralized asynchronous training on heterogeneous devices, arXiv preprint arXiv:2401.01728, (2024).

[62] Meta, The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, URL https://ai.meta.com/blog/llama-4-multimodal-intelligence, (2025).

[63] Minimax, Meet minimax-m2, URL https://huggingface.co/MiniMaxAI/MiniMax-M2, (2025).

[64] L. Morra, A. Biondo, N. Poerio, And F. Lamberti, Mixo: Mixture of experts-based visual odometry for multicamera autonomous systems, IEEE Transactions on Consumer Electronics, 2023, pp. 261–270.

[65] S. Mu And S. Lin, A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications, arXiv preprint arXiv:2503.07137, (2025).

[66] B. Mustafa, C. Riquelme, J. Puigcerver, R. Jenatton, And N. Houlsby, Multimodal contrastive learning with limoe: the language-image mixture of experts, Advances in Neural Information Processing Systems, 2022, pp. 9564–9576.

[67] D. Narayanan, A. Harlap, A. Phanishayee, V. Seshadri, N. R. Devanur, G. R. Ganger, P. B. Gibbons, And M. Zaharia, Pipedream: Generalized pipeline parallelism for dnn training, in Proceedings of the 27th ACM symposium on operating systems principles, 2019, pp. 1–15.

[68] D. Narayanan, M. Shoeybi, J. Casper, P. Legresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, Et Al., Efficient large-scale language model training on gpu clusters using megatron-lm, in Proceedings of the international conference for high performance computing, networking, storage and analysis, 2021, pp. 1–15.

[69] X. Nie, P. Zhao, X. Miao, T. Zhao, And B. Cui, Hetumoe: An efficient trillion-scale mixture-of-expert distributed training system, arXiv preprint arXiv:2203.14685, (2022).

[70] S. Pavlitska, C. Hubschneider, L. Struppek, And J. M. Zöllner, Sparsely-gated mixture-of-expert layers for cnn interpretability, in 2023 International Joint Conference on Neural Networks (IJCNN), IEEE, 2023, pp. 1–10.

[71] Z. Peng, T. Wang, C. Zhao, G. Liao, Z. Lin, Y. Liu, B. Cao, L. Shi, Q. Yang, And S. Zhang, A survey of zero-knowledge proof based verifiable machine learning, arXiv preprint arXiv:2502.18535, (2025).

[72] S. Pini, C. S. Perone, A. Ahuja, A. S. R. Ferreira, M. Niendorf, And S. Zagoruyko, Safe real-world autonomous driving by learning to predict and plan with a mixture of experts, in 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 10069–10075.

[73] J. Qi, W. Zhu, L. Li, M. Wu, Y. Wu, W. He, X. Gao, J. Zeng, And M. Heinrich, Dilocox: A low-communication large-scale training framework for decentralized cluster, arXiv preprint arXiv:2506.21263, (2025).

[74] P. Qi, X. Wan, G. Huang, And M. Lin, Zero bubble (almost) pipeline parallelism, in The Twelfth International Conference on Learning Representations, 2024.

[75] Z. Qiu, Z. Huang, B. Zheng, K. Wen, Z. Wang, R. Men, I. Titov, D. Liu, J. Zhou, And J. Lin, Demons in the detail: On implementing load balancing loss for training specialized mixture-of-expert models, arXiv preprint arXiv:2501.11873, (2025).

[76] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Et Al., Improving language understanding by generative pre-training, OpenAI Blog, (2018).

[77] S. Rajbhandari, C. Li, Z. Yao, M. Zhang, R. Y. Aminabadi, A. A. Awan, J. Rasley, And Y. He, Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale, in International conference on machine learning, PMLR, 2022, pp. 18332–18346.

[78] S. Rajbhandari, J. Rasley, O. Ruwase, And Y. He, Zero: Memory optimizations toward training trillion parameter models, in SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2020, pp. 1–16.

[79] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. Susano Pinto, D. Keysers, And N. Houlsby, Scaling vision with sparse mixture of experts, Advances in Neural Information Processing Systems, 2021, pp. 8583–8595.

[80] M. Ryabinin, T. Dettmers, M. Diskin, And A. Borzunov, Swarm parallelism: Training large models can be surprisingly communication-efficient, in International Conference on Machine Learning, PMLR, 2023, pp. 29416–29440.

[81] M. Ryabinin, E. Gorbunov, V. Plokhotnyuk, And G. Pekhimenko, Moshpit sgd: Communication-efficient decentralized training on heterogeneous unreliable devices, Advances in Neural Information Processing Systems, 2021, pp. 18195–18211.

[82] M. Ryabinin And A. Gusev, Towards crowdsourced training of large neural networks using decentralized mixture-of-experts, Advances in Neural Information Processing Systems, 2020, pp. 3659–3672.

[83] A. Sergeev And M. Del Balso, Horovod: fast and easy distributed deep learning in tensorflow, arXiv preprint arXiv:1802.05799, (2018).

[84] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, And J. Dean, Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, arXiv preprint arXiv:1701.06538, (2017).

[85] M. Shoeybi, M. Patwary, R. Puri, P. Legresley, J. Casper, And B. Catanzaro, Megatron-lm: Training multi-billion parameter language models using model parallelism, arXiv preprint arXiv:1909.08053, (2019).

[86] S. Singh, O. Ruwase, A. A. Awan, S. Rajbhandari, Y. He, And A. Bhatele, A hybrid tensor-expert-data parallelism approach to optimize mixture-of-experts training, in Proceedings of the 37th International Conference on Supercomputing, 2023, pp. 203–214.

[87] Q. Sun, H. Wang, J. Zhan, F. Nie, X. Wen, L. Xu, K. Zhan, P. Jia, X. Lang, And H. Zhao, Generalizing motion planners with mixture of experts for autonomous driving, arXiv preprint arXiv:2410.15774, (2024).

[88] S. Sun, R. Wang, And B. An, Quantitative stock investment by routing uncertainty-aware trading experts: A multi-task learning approach, arXiv preprint arXiv:2207.07578, (2022).

[89] Y. Tang, X. Li, F. Liu, W. Guo, H. Zhou, Y. Wang, K. Han, X. Yu, J. Li, H. Zang, Et Al., Pangu pro moe: Mixture of grouped experts for efficient sparsity, arXiv preprint arXiv:2505.21411, (2025).

[90] Y. Tang, Y. Yin, Y. Wang, H. Zhou, Y. Pan, W. Guo, Z. Zhang, M. Rang, F. Liu, N. Zhang, Et Al., Pangu ultra moe: How to train your big moe on ascend npus, arXiv preprint arXiv:2505.04519, (2025).

[91] Z. Tang, Y. Wang, X. He, L. Zhang, X. Pan, Q. Wang, R. Zeng, K. Zhao, S. Shi, B. He, Et Al., Fusionai: Decentralized training and deploying llms with massive consumer-level gpus, arXiv preprint arXiv:2309.01172, (2023).

[92] K. Team, Kimi k2: Open agentic intelligence, arXiv preprint arXiv:2507.20534, (2025).

[93] Q. Team, Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters, URL https://qwenlm.github.io/blog/qwen-moe, (2024).

[94] T. M. R. Team, Introducing dbrx: A new state-of-the-art open llm, URL https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm, (2024).

[95] D. Vallarino, A dynamic approach to stock price prediction: Comparing rnn and mixture of experts models across different volatility profiles, arXiv preprint arXiv:2410.07234, (2024).

[96] D. Vallarino, How do consumers really choose: Exposing hidden preferences with the mixture of experts model, arXiv preprint arXiv:2503.05800, (2025).

[97] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, And I. Polosukhin, Attention is all you need, Advances in neural information processing systems, 2017, pp. 6000 - 6010.

[98] A. Wang, X. Sun, R. Xie, S. Li, J. Zhu, Z. Yang, P. Zhao, J. Han, Z. Kang, D. Wang, Et Al., Hmoe: Heterogeneous mixture of experts for language modeling, arXiv preprint arXiv:2408.10681, (2024).

[99] B. Wang, Q. Xu, Z. Bian, And Y. You, 2.5-dimensional distributed model training, CoRR, (2021).

[100] J. Wang, X. Yang, Z. Wang, X. Wei, A. Wang, D. He, And K. Wu, Efficient mixture-of-expert for video-based driver state and physiological multi-task estimation in conditional autonomous driving, arXiv preprint arXiv:2410.21086, (2024).

[101] S. Wang, G. He, G.-W. Kim, Y. Zhou, And S. J. Park, Toward cost-efficient serving of mixture-of-experts with asynchrony, arXiv preprint arXiv:2505.08944, (2025).

[102] X. Wang, F. Yu, L. Dunlap, Y.-A. Ma, R. Wang, A. Mirhoseini, T. Darrell, And J. E. Gonzalez, Deep mixture of experts via shallow embedding, in Uncertainty in artificial intelligence, PMLR, 2020, pp. 552–562.

[103] T. Wei, B. Zhu, L. Zhao, C. Cheng, B. Li, W. Lü, P. Cheng, J. Zhang, X. Zhang, L. Zeng, X. Wang, Y. Ma, R. Hu, S. Yan, H. Fang, And Y. Zhou, Skywork-moe: A deep dive into training techniques for mixture-of-experts language models, arXiv preprint arXiv:2406.06563, (2024).

[104] Z. Wei, D. Chen, Y. Zhang, D. Wen, X. Nie, And L. Xie, Deep reinforcement learning portfolio model based on mixture of experts, Applied Intelligence, 2025.

[105] S. Wellington, Basedai: A decentralized p2p network for zero knowledge large language models (zk-llms), arXiv preprint arXiv:2403.01008, (2024).

[106] X. Wu, J. Rao, And W. Chen, Atom: Asynchronous training of massive models for deep learning in a decentralized environment, arXiv preprint arXiv:2403.10504, (2024).

[107] Y. Wu, X. Liu, S. Jin, C. Xu, F. Qian, Z. M. Mao, M. Lentz, D. Zhuo, And I. Stoica, Hetermoe: Efficient training of mixture-of-experts models on heterogeneous gpus, arXiv preprint arXiv:2504.03871, (2025).

[108] Z. Wu, X. Chen, Z. Pan, X. Liu, W. Liu, D. Dai, H. Gao, Y. Ma, C. Wu, B. Wang, Et Al., Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding, arXiv preprint arXiv:2412.10302, (2024).

[109] K. Xiao, X. Jiang, P. Hou, And H. Zhu, Autoeis: Automatic feature embedding, interaction and selection on default prediction, Information Processing & Management, 2024.

[110] L. Xie, T. Luan, W. Cai, G. Yan, Z. Chen, N. Xi, Y. Fang, Q. Shen, Z. Wu, And J. Yuan, dflmoe: Decentralized federated learning via mixture of experts for medical data analysis, in Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 10203–10213.

[111] Z. Xing, Z. Zhang, Z. Zhang, Z. Li, M. Li, J. Liu, Z. Zhang, Y. Zhao, Q. Sun, L. Zhu, Et Al., Zero-knowledge proof-based verifiable decentralized machine learning in communication network: A comprehensive survey, IEEE Communications Surveys & Tutorials, 2025.

[112] Q. Xu And Y. You, An efficient 2d method for training super-large deep learning models, in 2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2023, pp. 222–232.

[113] X. Xu, L. Kong, H. Shuai, L. Pan, Z. Liu, And Q. Liu, Limoe: Mixture of lidar representation learners from automotive scenes, in Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 27368–27379.

[114] Y. Xu, J. Wang, R. Zhang, C. Zhao, D. Niyato, J. Kang, Z. Xiong, B. Qian, H. Zhou, S. Mao, Et Al., Decentralization of generative ai via mixture of experts for wireless networks: A comprehensive survey, arXiv preprint arXiv:2504.19660, (2025).

[115] F. Xue, Z. Zheng, Y. Fu, J. Ni, Z. Zheng, W. Zhou, And Y. You, Openmoe: An early effort on open mixture-of-experts language models, arXiv preprint arXiv:2402.01739, (2024).

[116] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, Et Al., Qwen3 technical report, arXiv preprint arXiv:2505.09388, (2025).

[117] Z. Yang, Y. Chai, X. Jia, Q. Li, Y. Shao, X. Zhu, H. Su, And J. Yan, Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving, arXiv preprint arXiv:2505.16278, (2025).

[118] L. Yu, S. Cheng, Z. Huang, J. Zhang, C. Shen, J. Lu, L. Yang, F. Zhang, And J. Ma, Sael: Leveraging large language models with adaptive mixture-of-experts for smart contract vulnerability detection, arXiv preprint arXiv:2507.22371, (2025).

[119] Y. Yu And P. Tiwari, Finteamexperts: Role specialized moes for financial analysis, arXiv preprint arXiv:2410.21338, (2024).

[120] H. Yuan, L. Yu, Z. Huang, J. Zhang, J. Lu, S. Cheng, L. Yang, F. Zhang, J. Ma, And C. Zuo, Mos: Towards effective smart contract vulnerability detection through mixture-of-experts tuning of large language models, arXiv preprint arXiv:2504.12234, (2025).

[121] X. Yuan, W. Kong, Z. Luo, And M. Xu, Efficient inference offloading for mixture-of-experts large language models in internet of medical things, Electronics, 2024.

[122] S. E. Yuksel, J. N. Wilson, And P. D. Gader, Twenty years of mixture of experts, IEEE transactions on neural networks and learning systems, 2012, pp. 1177–1193.

[123] M. Zhai, J. He, Z. Ma, Z. Zong, R. Zhang, And J. Zhai, {SmartMoE}: Efficiently training {Sparsely-Activated} models through combining offline and online parallelization, in 2023 USENIX Annual Technical Conference (USENIX ATC 23), 2023, pp. 961–975.

[124] H. Zhang, Y. Zhao, C. Angione, H. Yang, J. Buban, A. Farhan, F. Johnston, And P. Colangelo, Towards secure and private ai: A framework for decentralized inference, arXiv preprint arXiv:2407.19401, (2024).

[125] J. Zhang, X. Qu, T. Zhu, And Y. Cheng, Clip-moe: Towards building mixture of experts for clip with diversified multiplet upcycling, arXiv preprint arXiv:2409.19291, (2024).

[126] Z. Y. Zhang, B. Ding, And B. K. H. Low, Memory-efficient and privacy-preserving collaborative training for mixture-of-experts llms, arXiv preprint arXiv:2506.02965, (2025).

[127] C. Zhao, S. Zhou, L. Zhang, C. Deng, Z. Xu, Y. Liu, K. Yu, J. Li, And L. Zhao, deepep: an efficient expert-parallel communication library. URL https://github.com/deepseek-ai/, (2025).

[128] Y. Zhou, T. Lei, H. Liu, N. Du, Y. Huang, V. Zhao, A. M. Dai, Q. V. Le, J. Laudon, Et Al., Mixture-of-experts with expert choice routing, Advances in Neural Information Processing Systems, 2022, pp. 7103–7114.

[129] B. Zoph, I. Bello, S. Kumar, N. Du, Y. Huang, J. Dean, N. Shazeer, And W. Fedus, St-moe: Designing stable and transferable sparse expert models, arXiv preprint arXiv:2202.08906, (2022).