From Centralized to Composable: Advances in Distributed and Multimodal Language Modeling

Irma Mirta[a], Klaus Elli[a]

[a] *University of Stuttgart, Germany*

Abstract

The advent of large language models (LLMs) has ushered in a new era of general-purpose artificial intelligence capable of language understanding, generation, and reasoning. Recent progress has further extended these capabilities to the multimodal domain, where models integrate textual, visual, auditory, and sensory information. As the scale and complexity of both unimodal and multimodal LLMs grow, centralized training and inference become increasingly infeasible due to computational, memory, energy, and privacy constraints. Distributed architectures—spanning data parallelism, model parallelism, pipeline sharding, federated learning, and expert specialization—have emerged as necessary frameworks for scalable deployment and collaboration across modalities and domains.

This survey provides a comprehensive review of the advances, methodologies, and challenges in distributed LLMs and multimodal large language models (MLLMs). We begin with a mathematical characterization of distributed LLM architectures, followed by a taxonomy of communication-efficient fusion mechanisms, parallel training strategies, and alignment techniques across modalities. We discuss theoretical underpinnings of multimodal representation learning, including universal approximation, modality-aware attention, and conditional computation. We then examine training paradigms across decentralized and federated infrastructures, emphasizing optimization under partial supervision, low-bandwidth constraints, and heterogeneity in hardware and data distribution.

The survey further explores real-world applications, deployment architec-

tures, and constraints on inference latency, energy efficiency, and privacy. We highlight deployment patterns ranging from cloud-centric to fully edge-based systems and discuss model compression techniques including quantization, pruning, and mixture-of-experts routing. The final sections identify open research challenges in scalability, cross-modal generalization, robustness to missing or adversarial modalities, and the interpretability and safety of distributed MLLMs. We conclude with future directions that emphasize sustainability, democratization, and theoretical convergence in multimodal distributed intelligence.

This survey aims to serve as a foundational resource for researchers and practitioners building the next generation of distributed, multimodal, and human-aligned AI systems.

---

1. Introduction

The past few years have witnessed a paradigm shift in the field of artificial intelligence, spearheaded by the rapid advancement of large language models (LLMs) such as GPT, PaLM, LLaMA, and others. These models, with hundreds of billions of parameters, have demonstrated unprecedented capabilities across a wide spectrum of natural language processing (NLP) tasks, including machine translation, summarization, dialogue systems, and code generation [1]. Simultaneously, the emergence of multimodal large language models (MLLMs) — systems capable of processing and reasoning across diverse modalities such as text, images, audio, and video — has further expanded the boundaries of what AI can achieve, enabling breakthroughs in vision-language tasks, text-to-image generation, and embodied AI applications. As these models continue to scale, the associated computational, architectural, and algorithmic challenges demand a reevaluation of traditional training, inference, and deployment paradigms. A key enabler of these advancements is the development of distributed training and inference systems. Training modern LLMs and MLLMs often involves the coordinated use of thousands of GPUs across multiple data centers, leveraging techniques such as data parallelism, model parallelism, pipeline parallelism, and

tensor parallelism. These distributed systems must address complex issues such as communication overhead, fault tolerance, memory efficiency, and workload balancing. At the same time, ensuring scalability and efficiency while maintaining model quality and minimizing training time is a non-trivial endeavor [2]. The intricacies of distributed training architectures, such as DeepSpeed, Megatron-LM, and Alpa, have become central to the progress of large-scale AI. The integration of multiple modalities in large-scale models introduces additional layers of complexity. Multimodal learning requires not only the design of architectures that can jointly process heterogeneous data but also the creation of aligned and high-quality datasets that span different domains [3]. This raises new questions regarding modality fusion strategies, cross-modal alignment, and representational learning. Moreover, ensuring the efficiency and scalability of multimodal models under distributed settings introduces unique bottlenecks, including the handling of modality-specific preprocessing, synchronized data loading, and heterogeneous compute resource allocation [4]. Despite the impressive performance of LLMs and MLLMs, significant challenges remain. Training these models is immensely resource-intensive, often costing millions of dollars and consuming vast amounts of energy, raising concerns about environmental sustainability and equitable access to AI technology. Moreover, the opacity and complexity of these models exacerbate issues of interpretability, fairness, and accountability [5]. Distributed settings further complicate these concerns, as coordination and reproducibility become harder to guarantee across geographically dispersed training nodes [6]. Additionally, the deployment of such models for real-time applications requires highly optimized inference pipelines that can meet latency and throughput constraints while preserving model performance [7]. The interplay between LLMs and MLLMs also highlights important research frontiers. On the one hand, the convergence of language and vision in foundational models like GPT-4, Gemini, and Flamingo underscores the need for unified architectures that can generalize across modalities. On the other hand, the distributed training of such models necessitates new methods for gradient synchronization, parameter sharding, and multimodal tokenization. Further-

more, new use cases — such as AI-powered agents that interact with humans via voice, text, and visual cues — demand models that are not only large and powerful but also efficient, adaptive, and trustworthy [8]. This survey aims to provide a comprehensive overview of the advances, challenges, and future directions in the field of distributed LLMs and multimodal large language models. We begin by categorizing the key components of modern LLM and MLLM architectures and exploring how they are adapted or redesigned for distributed settings. We then delve into the methodologies and systems that enable large-scale distributed training, including parallelism strategies, hardware acceleration, and systems optimizations. The survey also examines the specific challenges of multimodal learning at scale, from data curation to cross-modal integration, as well as the implications for downstream applications [9]. In addition, we explore the pressing issues related to the responsible development and deployment of these models, such as data privacy, bias mitigation, and energy efficiency. Finally, we outline emerging trends and open research directions, including the democratization of model training, advances in decentralized and federated learning, and the quest for models that are both general-purpose and domain-adaptive [10]. By synthesizing knowledge across these dimensions, this survey aims to serve as a valuable resource for researchers, practitioners, and policymakers seeking to navigate the complex landscape of large-scale, multimodal, and distributed AI systems. The future of artificial intelligence hinges not only on the capabilities of the models we build but also on the architectures, algorithms, and principles that govern their construction and use.

2. Architectures and Parallelism Strategies for Distributed LLMs and MLLMs

The design of large-scale language models and multimodal systems necessitates sophisticated architectural planning and training methodologies to ensure efficient scaling across distributed environments. Fundamentally, the training of an LLM or MLLM can be formalized as the minimization of a loss function

$\mathcal{L}(\theta)$ over model parameters $\theta \in \mathbb{R}^d$, where d is typically on the order of billions or trillions. The optimization is conducted over a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, which may include not only text pairs but also image-text tuples, audio clips, or even video frames in the case of MLLMs. Given the computational infeasibility of training such models on a single machine, distributed strategies have been developed to partition the workload both horizontally and vertically [11]. Let us denote the model as a composite function $f_\theta(x) = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}(x)$ where L represents the number of layers. Distributed training attempts to parallelize either the computation of each layer, the data passed through them, or the parameter updates. These strategies fall into three main categories: **data parallelism**, **model parallelism**, and **pipeline parallelism** [12]. Data parallelism is conceptually the simplest approach. Here, the training dataset is partitioned into K disjoint subsets $\mathcal{D}_1, \ldots, \mathcal{D}_K$ and distributed across K workers, each holding a replica of the full model. Gradients are computed independently across workers and synchronized at each step using an all-reduce operation:

$$\nabla \mathcal{L}(\theta) = \frac{1}{K} \sum_{k=1}^{K} \nabla \mathcal{L}_k(\theta).$$

While this approach is communication-heavy, especially for large models, it scales well when the batch size can be increased proportionally to the number of workers. Model parallelism, on the other hand, splits the model itself across multiple devices [13]. For example, in tensor model parallelism, large matrix multiplications in layers such as self-attention or feed-forward blocks are split along rows or columns [14]. This reduces memory load per device but introduces cross-device communication during each forward and backward pass. Formally, let $\theta = [\theta_1, \ldots, \theta_K]$ be partitioned across devices such that $f_\theta(x) = f_{\theta_K}^{(K)} \circ \cdots \circ f_{\theta_1}^{(1)}(x)$ and each function component is computed on a separate device. Pipeline parallelism exploits sequential layer execution by placing consecutive layers on different devices, enabling microbatch-based pipelining across forward and backward passes [15]. If we denote the batch size as B and split it into M microbatches, the training can be pipelined over $T = M + K - 1$ stages with K being the number of pipeline stages, leading to increased utilization but

requiring careful scheduling to avoid bubbles in the pipeline. Table 1 compares these strategies across key dimensions.

Table 1: Comparison of Parallelism Strategies in Distributed LLM and MLLM Training

Property	**Data Parallelism**	**Model Parallelism**	**Pipeline Parallelism**
Memory Efficiency	Low	High	Moderate
Communication Cost	High	Moderate	Low
Implementation Complexity	Low	High	High
Scalability	Good (with batch size)	Good (with model size)	Moderate
Inter-device Synchronization	Gradient All-reduce	Activation Transfer	Pipeline Scheduling

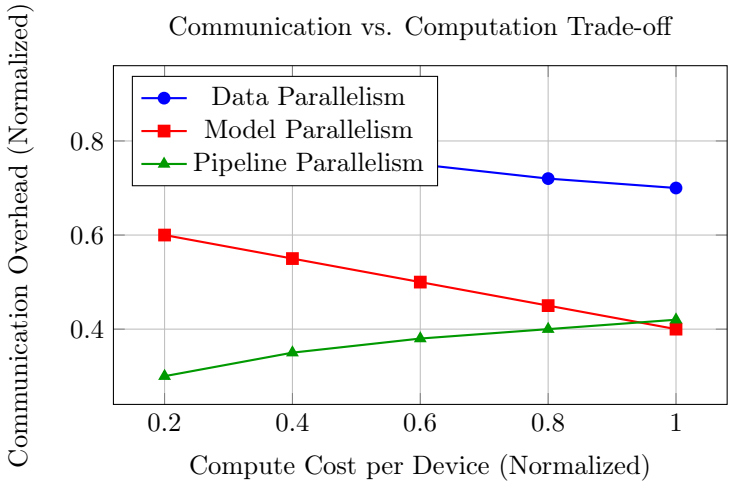

Figure 1: Trade-off curve between communication overhead and computation cost for various parallelism strategies [16]. Lower-left is ideal.

In multimodal settings, these strategies must be extended or adapted [17]. For example, in a vision-language model such as Flamingo, the model must process both image features (typically extracted by a vision transformer) and text tokens. Each modality may exhibit different compute and memory characteristics, requiring heterogeneous device assignments [18]. In such cases, hierarchical parallelism becomes essential, where data parallelism is applied at the batch level, while model and pipeline parallelism are jointly applied within

the transformer backbone and vision encoder. Furthermore, as models become increasingly modular — incorporating prompt tuning layers, retrieval modules, or external memory — parallelization strategies must account for the modular boundaries and their implications on data flow and latency. Formally, let the multimodal model be $f_\theta(x_{\text{text}}, x_{\text{image}}) = f_{\text{fuse}}(f_{\text{text}}(x_{\text{text}}), f_{\text{img}}(x_{\text{image}}))$, and assume each subcomponent resides on a separate compute group. The efficient training of such architectures demands overlapping computation with communication and may benefit from asynchronous gradient updates or sparsely activated subnetworks to reduce the active parameter count [19]. In summary, the choice and configuration of parallelism strategies in distributed LLMs and MLLMs is tightly coupled with the model architecture, resource availability, and task-specific demands. Understanding the mathematical structure and communication topology of these systems is critical for their efficient deployment. The next section will investigate the software frameworks and system-level optimizations that enable these strategies at scale [20].

3. System-Level Optimizations and Frameworks for Distributed LLM and MLLM Training

Training large-scale LLMs and multimodal models on distributed hardware involves not only algorithmic parallelism but also significant system-level engineering. Efficient scaling across nodes requires precise coordination of memory, compute, networking, and I/O resources, typically mediated by specialized training frameworks [21]. This section examines the fundamental system-level challenges encountered during large-scale training and surveys the major software solutions that abstract and optimize these challenges. Let us denote a training cluster as a tuple $\mathcal{C} = (N, G, L)$, where N is the number of compute nodes, G the GPU count per node, and L the logical topology (e.g., ring, fully connected, hierarchical) [22]. Each training iteration comprises three dominant phases: (1) forward and backward computation $\mathcal{F}_{\text{comp}}$, (2) gradient synchronization $\mathcal{F}_{\text{sync}}$, and (3) parameter update $\mathcal{F}_{\text{update}}$ [23]. The total training time

per step T_{step} can be approximated as:

$$T_{\text{step}} \approx \max\left(T_{\mathcal{F}_{\text{comp}}}, T_{\mathcal{F}_{\text{sync}}} + T_{\mathcal{F}_{\text{update}}}\right) + T_{\text{overhead}},$$

where T_{overhead} includes scheduling, memory management, and I/O latency [24]. System-level optimizations focus on minimizing T_{sync} and T_{overhead} through low-latency communication backbones, hierarchical memory caching, and overlapping computation with communication. For instance, NVLink, NVSwitch, and InfiniBand RDMA drastically reduce inter-GPU and inter-node communication time, especially when used with optimized collectives such as NCCL's ring-based all-reduce or hierarchical all-gather algorithms. Beyond hardware, distributed training frameworks such as Megatron-LM, DeepSpeed, Alpa, and FSDP (Fully Sharded Data Parallel) offer various trade-offs in performance, scalability, and usability [25]. These frameworks manage memory fragmentation, activation recomputation, kernel fusion, and sharding of optimizer states to fit large models into limited GPU memory. The total memory usage per device M_{total} during training can be expressed as:

$$M_{\text{total}} = M_{\text{params}} + M_{\text{activations}} + M_{\text{optimizer}} + M_{\text{temporary}},$$

where $M_{\text{params}} \propto \frac{|\theta|}{P}$ for P-way model sharding, and $M_{\text{optimizer}}$ typically requires $2\times$ or $3\times$ the parameter memory for optimizers like Adam. Optimizing M_{total} is crucial to avoid out-of-memory (OOM) errors during large batch or sequence length training [26].

Table 2: Feature Comparison of Distributed Training Frameworks

Framework	**Optimizer Sharding**	**Activation Checkpointing**	**Parallelism Support**	**Mixed**
DeepSpeed	Yes (ZeRO 1/2/3)	Yes	Data, Pipeline, Model	Yes (FP
Megatron-LM	Partial	Manual	Tensor, Pipeline	
Alpa	Yes	Yes	Fully Automatic (XLA)	
FSDP	Yes (per-layer)	Yes	Data + Sharded Model	
Colossal-AI	Yes	Yes	3D Parallelism	

Frameworks like DeepSpeed employ memory-centric strategies such as ZeRO

(Zero Redundancy Optimizer), which partitions gradients, optimizer states, and parameters across devices, effectively reducing M_{params} and $M_{\text{optimizer}}$ to $\mathcal{O}(1/P)$ per GPU [27]. When combined with offloading techniques — e.g., ZeRO-Offload — model state can be moved to CPU or NVMe, albeit at the cost of increased latency. Alpa adopts a different strategy based on automated parallelization using XLA (Accelerated Linear Algebra), which partitions computation graphs based on cost models [28]. It formulates the parallelization problem as a combinatorial optimization problem over a parallel execution plan $\mathcal{P}^* = \arg\min_{\mathcal{P}} \text{Cost}(\mathcal{P})$, where the cost includes communication volume, device utilization, and memory pressure. This approach generalizes well for transformer-based MLLMs, especially in research workflows where minimal manual tuning is desirable [29]. FSDP in PyTorch offers fine-grained per-layer sharding of parameters and gradients, making it suitable for scenarios involving memory-constrained GPUs [30]. It performs forward computation with sharded parameters and reconstructs full tensors only when needed, minimizing peak memory [31]. The trade-off is additional synchronization and memory bandwidth usage. Moreover, communication-aware graph schedulers and tensor rematerialization (checkpointing) help balance compute and memory. Let $\mathcal{G} = (V, E)$ be a computation graph of model operations with associated memory cost $m(v)$ and compute cost $c(v)$. Checkpointing can be framed as a partitioning problem to select a subset $S \subset V$ such that the recompute cost $\sum_{v \notin S} c(v)$ is minimized subject to a memory constraint $\sum_{v \in S} m(v) \leq M_{\text{budget}}$ [32]. Another emerging trend is the integration of hardware accelerators such as TPUs and AI-specific chips (e.g., Cerebras, Groq) into the training pipeline. These systems optimize dense matrix operations and exhibit high memory bandwidth, reducing the step time T_{step} significantly. However, they require customized compilers and kernel optimizations, making them less accessible outside large organizations. Finally, MLLMs introduce unique challenges due to modality-specific data prefetching, preprocessing, and augmentation [33]. For instance, vision encoders require image decoding and resizing, which can become a bottleneck. Hence, frameworks must integrate asynchronous I/O pipelines and hardware-

accelerated preprocessing (e.g., NVIDIA DALI) to keep GPUs saturated [34]. Let T_{input} denote the input pipeline latency; for optimal throughput, one must ensure $T_{\text{input}} < T_{\text{step}} - T_{\text{comp}}$ [35]. In conclusion, distributed LLM and MLLM training hinges not just on model architecture or algorithmic design, but equally on the systems stack that supports their training. Understanding the interplay between memory management, compute scheduling, and interconnect performance is essential for building scalable, high-performance AI systems. The next section will explore the specialized considerations and innovations required for training multimodal foundation models at scale.

4. Training Multimodal Large Language Models at Scale: Modal Alignment, Fusion Strategies, and Distributed Constraints

Multimodal Large Language Models (MLLMs) extend traditional LLMs by integrating diverse input modalities such as images, video, audio, and structured metadata, necessitating sophisticated architectures capable of unified representation learning [36]. These models, denoted as functions $f_\theta(x^{(1)}, x^{(2)}, \ldots, x^{(m)})$ where each $x^{(i)}$ represents an input from modality i, must learn a joint embedding or reasoning space that preserves modality-specific semantics while enabling cross-modal interaction [37].

Cross-Modal Alignment and Representation Fusion

The central task in multimodal modeling is alignment: mapping diverse modalities into a shared latent space $\mathcal{Z}$ such that semantically related pairs across modalities (e.g., image-caption, video-transcript) lie close together under some distance metric $d : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}^+$. Formally, we desire:

$$d\left(f_\theta^{\text{img}}(x^{(\text{img})}), f_\theta^{\text{txt}}(x^{(\text{txt})})\right) \ll d\left(f_\theta^{\text{img}}(x^{(\text{img})}), f_\theta^{\text{txt}}(\tilde{x}^{(\text{txt})})\right),$$

for positive image-text pairs $(x^{(\text{img})}, x^{(\text{txt})})$ and negative pairs $(x^{(\text{img})}, \tilde{x}^{(\text{txt})})$ [38]. Contrastive learning frameworks (e.g., CLIP, ALIGN) implement this via

a symmetric InfoNCE loss:

$$\mathcal{L}_{\text{contrast}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\langle z_i^{(\text{img})}, z_i^{(\text{txt})}\rangle/\tau)}{\sum_{j=1}^{N}\exp(\langle z_i^{(\text{img})}, z_j^{(\text{txt})}\rangle/\tau)},$$

where $z^{(\cdot)} = f_\theta^{(\cdot)}(x^{(\cdot)})$ are the encoded representations and τ is a learned temperature parameter. Alternatively, fusion-based MLLMs employ attention mechanisms across modality-specific tokens, particularly in encoder-decoder or autoregressive settings. Denote the joint input as a sequence of token embeddings:

$$\mathbf{X} = [\mathbf{x}_1^{(\text{txt})}, \ldots, \mathbf{x}_{T_1}^{(\text{txt})}, \mathbf{x}_1^{(\text{img})}, \ldots, \mathbf{x}_{T_2}^{(\text{img})}],$$

where positional and modality embeddings are added before passing into a transformer. Cross-attention layers compute modality-bridging dependencies:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V,$$

where $Q = XW_Q$, $K = XW_K$, and $V = XW_V$ include tokens from multiple modalities [39]. The structure of attention masks and key-query modality segregation impacts both expressivity and memory cost [40].

Distributed Training Challenges in Multimodal Pipelines

Scaling MLLMs introduces unique bottlenecks that are not observed in purely textual LLMs [41]. First, data loading becomes heterogeneous: image preprocessing (JPEG decoding, resizing), audio processing (STFT, MFCCs), and video processing (frame sampling, interpolation) each introduce distinct latencies $T_{\text{load}}^{(i)}$ per modality. These must be synchronized or buffered in distributed pipelines to prevent GPU underutilization:

$$T_{\text{idle}}^{(j)} = \max_i T_{\text{load}}^{(i)} - T_{\text{load}}^{(j)}.$$

Second, the memory footprint per example varies significantly [42]. For instance, an image-token sequence may be $10\times$ longer than a sentence, and attention complexity scales quadratically: $\mathcal{O}(T^2 \cdot d)$, where T is sequence length and d embedding dimension. This leads to per-GPU memory pressure:

$$M_{\text{attn}} \propto \sum_{i=1}^{m} T_i^2 \cdot d.$$

To manage this, distributed training uses modality-aware batch construction, e.g., padding and bucketing techniques that group samples of similar length and modality mix [43]. In cases with highly sparse modality presence (e.g., 70% text-only, 20% image-text, 10% video-text), a heterogeneous compute group assignment is employed, where specific GPU sets are reserved for high-cost samples [44]. Formally, define a routing function $r : \mathcal{B} \to \mathcal{C}$ from minibatches to compute clusters that minimizes expected step time:

$$r^* = \arg\min_r \mathbb{E}_{b\sim\mathcal{B}} \left[T_{\text{step}}(r(b))\right].$$

Modality Tokenization and Fusion Architectures

Tokenization of non-text modalities is non-trivial. For example, images are commonly tokenized using ViT-like patch embeddings $x^{(\text{img})} \mapsto \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ where each $\mathbf{x}_i \in \mathbb{R}^d$ represents a flattened 16×16 patch [45]. Audio and video may use convolutional frontends or spectrogram embeddings. The granularity of these tokens affects model depth and attention scaling, making hierarchical token reduction strategies such as Perceiver or CoAtNet necessary. Fusion timing — early, mid, or late — also plays a critical role in MLLM effectiveness [46]. Early fusion concatenates all modalities at input; mid-fusion aligns them in a shared encoder (e.g., cross-attention); and late fusion defers interaction until high-level representations are formed. Each comes with trade-offs:

- **Early Fusion:** maximizes low-level interaction, high memory cost.
- **Mid Fusion:** balances expressivity and efficiency, common in current models.
- **Late Fusion:** low computational overhead, but weak intermodal reasoning [47].

In recent architectures such as Flamingo, the attention mechanism is gated to condition text generation on vision tokens. In unified models like GPT-4V, modality routers dynamically switch input pathways at inference time. These mechanisms require additional design overhead in distributed settings, particularly for managing parameter sharding and attention mask consistency.

5. Communication Efficiency, Compression Techniques, and Memory Optimization in Distributed Multimodal LLMs

Training distributed multimodal large language models (MLLMs) at scale requires significant communication between GPUs and nodes, particularly during gradient synchronization and parameter updates. With growing model sizes and sequence lengths across modalities, the communication-to-computation ratio becomes a primary bottleneck, especially for attention-heavy architectures and dense transformer layers [48]. In this section, we formalize these challenges and review advanced compression and memory optimization strategies employed to mitigate them [49].

Communication Cost Models and Bottlenecks

In data-parallel training, the synchronization of gradients or parameters between devices introduces an all-reduce communication pattern [50]. For a model with $|\theta|$ parameters, and P parallel workers, the cost of all-reduce can be estimated as:

$$T_{\text{comm}} = \alpha \cdot \log P + \beta \cdot \frac{|\theta|}{P},$$

where α is the startup latency per communication round and β is the inverse bandwidth (time per byte) [51]. This is exacerbated in multimodal settings where $|\theta|$ often includes modality-specific branches, and the effective tensor size increases due to longer sequences or dense representations (e.g., visual tokens). To alleviate bandwidth pressure, model developers adopt low-precision communication protocols. Assuming full-precision gradients $\mathbf{g} \in \mathbb{R}^d$, a quantized form $\hat{\mathbf{g}}$ using b-bit precision reduces transmission size by a factor of $32/b$. However, naively applying quantization introduces numerical instability. Thus, techniques such as stochastic rounding or error compensation are used:

$$\mathbf{e}_{t+1} = \mathbf{g}_t - Q(\mathbf{g}_t + \mathbf{e}_t),$$

where $Q(\cdot)$ is the quantization operator and $\mathbf{e}_t$ the accumulated quantization error. This mechanism preserves convergence properties under appropriate assumptions on Q's unbiasedness.

Gradient Sparsification and Compression

An alternative to quantization is gradient sparsification, where only the top-k elements of each gradient vector are communicated. Denote $\text{Top}_k(\mathbf{g})$ as the sparse mask selecting the largest k elements in magnitude:

$$\hat{\mathbf{g}} = \mathbf{g} \odot \text{Top}_k(\mathbf{g}), \quad \text{with} \quad \|\hat{\mathbf{g}}\|_0 = k[52].$$

This reduces communication by $\mathcal{O}(d/k)$ but may degrade convergence unless compensated by techniques such as momentum correction or residual accumulation.

Memory-Constrained Training: Checkpointing and Offloading

Multimodal models often exceed GPU memory budgets due to the storage of activations for all modalities during backpropagation. Given an activation tensor $\mathbf{A} \in \mathbb{R}^{T \times d}$, storing $\mathcal{O}(L)$ such tensors for each layer incurs a total memory usage:

$$M_{\text{activations}} = L \cdot T \cdot d \cdot \text{sizeof(dtype)}[53].$$

Activation checkpointing reduces memory by recomputing intermediate states during backward pass. Let $\mathcal{C}$ be the set of checkpointed layers, then the recompute overhead is:

$$T_{\text{recompute}} = \sum_{\ell \in \mathcal{C}} c_\ell,$$

where c_ℓ is the compute time for layer ℓ. This technique trades compute for memory and is optimized via dynamic programming or heuristics based on memory budget constraints. For models exceeding even recomputed budgets, parameter and optimizer state offloading to CPU or NVMe is required. The offload latency T_{offload} is mitigated using overlap strategies:

$$T_{\text{effective}} = \max(T_{\text{compute}}, T_{\text{offload}} - T_{\text{overlap}}).$$

Unified Frameworks for Memory and Communication Optimization

Several training frameworks integrate these strategies natively. For example:

- **ZeRO-Infinity** supports NVMe offloading of optimizer and parameter states with memory-aware partitioning [54].
- **FSDP** shards parameters across GPUs, reducing peak memory to $\mathcal{O}(1/P)$, and supports mixed precision [55].
- **Colossal-AI** includes 3D parallelism combining tensor, pipeline, and sequence sharding for MLLMs.

Table 3: Overview of Optimization Techniques Across Frameworks

Framework	**Quantization**	**Checkpointing**	**Offloading**	**Gradient Sparsity**
DeepSpeed ZeRO-Infinity	FP8 / INT8	Yes	CPU + NVMe	Planned
FSDP (PyTorch)	FP16 / BF16	Yes	Partial	No
Colossal-AI	FP8 / INT4	Yes	CPU	Yes
FairScale	FP16	Manual	No	No

Case Study: Impact of Compression on Throughput

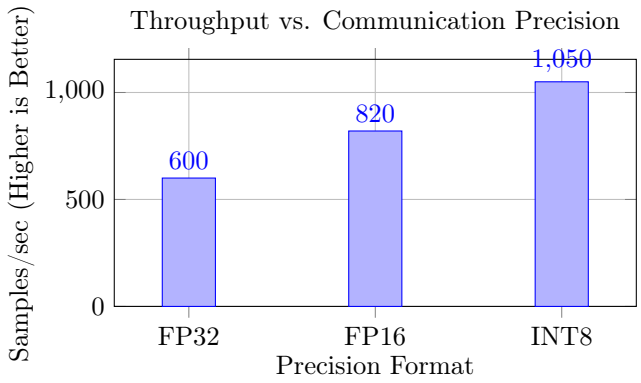

Figure 2: Effect of communication quantization on training throughput. Lower precision formats reduce gradient synchronization cost and improve throughput.

Figure 2 demonstrates the training throughput gain under various communication precisions. While FP16 is now standard in most frameworks, INT8 and

FP8 offer significant bandwidth savings without notable accuracy degradation when applied selectively to communication paths [56].

Toward Future Memory Hierarchies and In-Network Compute

As model sizes continue to grow beyond trillion-parameter scale [57] and input modalities proliferate, emerging architectures explore hierarchical memory systems (HBM + DRAM + NVMe) and programmable interconnects (e.g., in-network computation using SHARP or BlueField DPUs). Let $\mathcal{M} = \{M_1, M_2, \ldots, M_n\}$ represent the memory tiers with latency and bandwidth constraints (ℓ_i, b_i) [58]. An optimal parameter placement $\pi^* : \theta \to \mathcal{M}$ solves:

$$\pi^* = \arg\min_{\pi} \sum_{\theta_i} \frac{s(\theta_i)}{b_{\pi(\theta_i)}} + \ell_{\pi(\theta_i)},$$

subject to capacity constraints $\sum_{\theta_i \in M_j} s(\theta_i) \leq C_j$ [59]. In-network compute reduces T_{comm} by performing operations like all-reduce or quantization directly within network switches or DPUs, eliminating CPU/GPU intervention [60]. While still nascent, these technologies may define the next frontier in exascale MLLM training.

6. Evaluation, Generalization, and Safety of Distributed Multimodal LLMs

Evaluating the performance and trustworthiness of distributed multimodal large language models (MLLMs) is a non-trivial task, especially as these systems operate in high-dimensional spaces and must reason over heterogeneous data. The challenges lie not only in measuring accuracy but also in ensuring robust generalization across modalities, fair alignment with human values, and safe behavior in deployment. This section explores theoretical and empirical evaluation protocols, generalization bounds, and safety verification methods in distributed MLLM systems [61].

Cross-Modal Evaluation Metrics and Benchmarks

Let $\mathcal{X}_1, \ldots, \mathcal{X}_m$ denote input modalities (e.g., text, image, audio, video) and $\mathcal{Y}$ the output domain (e.g., language tokens, class labels) [62]. For a model $f_\theta : \mathcal{X}_1 \times \cdots \times \mathcal{X}_m \to \mathcal{Y}$, standard evaluation requires datasets $\mathcal{D}_{\text{test}}^{(i)}$ per task i with associated metrics $\mathcal{M}_i$ [63]. Common tasks include:

- **VQA:** Visual Question Answering (e.g., GQA, VizWiz) using accuracy, BLEU, CIDEr [64].
- **Image Captioning:** Evaluation via SPICE, METEOR, and BERTScore.
- **Text-to-Image Retrieval:** Using Recall@k, Mean Reciprocal Rank (MRR).
- **Multimodal Reasoning:** Tasks like ScienceQA or MMMU using exact match and explanation accuracy [65].

Let $\mathcal{L}_{\text{task}}^{(i)}$ be the task-specific loss. The overall performance metric in multi-objective evaluation is given by a weighted sum:

$$\mathcal{M}_{\text{global}} = \sum_{i=1}^{n} w_i \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{test}}^{(i)}}[\mathcal{M}_i(f_\theta(x), y)],$$

with weights w_i reflecting task relevance or data coverage.

Generalization Bounds in Cross-Modal and Distributed Contexts

From a theoretical perspective, multimodal generalization extends classical VC-dimension and Rademacher complexity to product spaces. Let $\mathcal{H}$ denote the hypothesis class of multimodal functions $f : \prod_{i=1}^{m} \mathcal{X}_i \to \mathbb{R}$, and $\mathcal{R}_n(\mathcal{H})$ its Rademacher complexity over n samples. For independent modality encoders ϕ_i and fusion function g, the composite function class $f(x) = g(\phi_1(x_1), \ldots, \phi_m(x_m))$ admits the bound:

$$\mathcal{R}_n(f) \leq \sum_{i=1}^{m} \mathcal{R}_n(\phi_i) + \mathcal{R}_n(g)[66].$$

In distributed settings with partitioned data, let $\mathcal{D}_p^{(j)}$ be the local distribution seen by node j. Heterogeneity (non-IID data) introduces drift $\delta_j =$

$\|\mathbb{E}_{x\sim\mathcal{D}_p^{(j)}}[\phi(x)] - \mathbb{E}_{x\sim\mathcal{D}}[\phi(x)]\|$, which affects the local empirical risk:

$$|\mathbb{E}_{\mathcal{D}_p^{(j)}}[\mathcal{L}(f(x), y)] - \mathbb{E}_{\mathcal{D}}[\mathcal{L}(f(x), y)]| \leq L_{\mathcal{L}} \cdot \delta_j.$$

Here, $L_{\mathcal{L}}$ is the Lipschitz constant of the loss function. This emphasizes the importance of federated averaging schemes and regularization for inter-node representation consistency.

Safety, Alignment, and Multimodal Robustness

Safety in MLLMs encompasses adversarial robustness, bias mitigation, toxic generation filtering, and multimodal grounding fidelity. Adversarial attacks may involve perturbations in any modality:

$$x' = x + \epsilon, \quad \text{such that} \quad \|\epsilon\| < \delta \quad \text{and} \quad f_\theta(x') \neq f_\theta(x)[67].$$

While textual adversaries often rely on synonym substitution or prompt injection, visual attacks (e.g., patch attacks or style transfer) exploit the non-linearity of visual encoders [68]. Defending against these requires certified robustness methods (e.g., randomized smoothing) and data augmentation pipelines with adaptive mixing strategies [69]. Bias detection is performed via demographic parity gaps, where for a protected attribute $a \in \mathcal{A}$:

$$\text{DPG}(a) = |\mathbb{P}(f_\theta(x) = y \mid A = a_1) - \mathbb{P}(f_\theta(x) = y \mid A = a_2)|.$$

Mitigation approaches include conditional training, counterfactual data augmentation, and model editing techniques such as ROME and MEMIT for localized intervention [70]. Multimodal alignment safety additionally involves "hallucination control"—ensuring the model does not generate unsupported claims based on partial modality inputs [71]. Let $\hat{y} = f(x^{(\text{text})}, x^{(\text{img})})$ be the output. If $x^{(\text{img})}$ is occluded or replaced with a blank image, robust grounding implies:

$$D(f(x^{(\text{text})}, x^{(\text{img})}), f(x^{(\text{text})}, \varnothing)) > \tau,$$

where $D(\cdot, \cdot)$ is a semantic divergence metric and τ is a threshold indicating dependency [72]. Factual inconsistency scores using models like GPTScore or UniEval are increasingly applied to enforce grounded generation.

Auditing and Evaluation at Scale

Safety and generalization in distributed MLLMs must be evaluated continuously at scale [73]. Auditing pipelines incorporate adversarial example generation, out-of-distribution (OOD) detection, and stress tests under perturbations. Metrics include:

- **Robust Accuracy:** Accuracy under distributional shift or noise.
- **Consistency Score:** Agreement across modalities, measured as:

$$\mathcal{C}(x^{(i)}, x^{(j)}) = \text{sim}(f(x^{(i)}), f(x^{(j)})),$$

  where sim is cosine or dot-product similarity.
- **Toxicity Score:** Measured using classifiers like Detoxify or Perspective API.
- **Bias Score:** From datasets like StereoSet or BOLD, capturing social stereotype frequency.

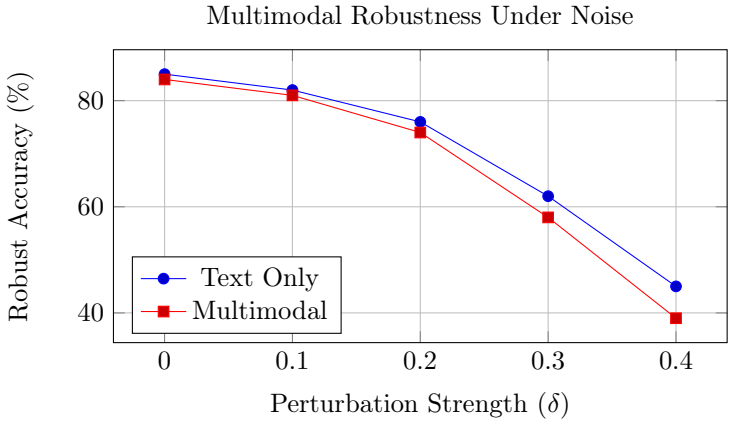

Figure 3: Comparison of robust accuracy under Gaussian noise perturbations [74]. Multimodal models show steeper degradation, highlighting modality-dependent fragility.

Federated Alignment and Decentralized Safety Protocols

When MLLMs are trained or deployed in federated or decentralized settings (e.g., mobile vision-text agents), centralized safety enforcement is impractical. Federated safety protocols rely on local detectors, differential privacy (ϵ-DP guarantees), and secure aggregation. Let f_j be the local model on node j with privatized updates Δ_j:

$$\Delta_j = \mathcal{Q}(f_j - f_{\text{global}}) + \mathcal{N}(0, \sigma^2 I),$$

where $\mathcal{Q}$ is a quantizer and $\mathcal{N}$ Gaussian noise added for DP guarantees. Global alignment is achieved via constrained optimization to ensure Δ_j satisfies fairness and robustness constraints encoded as Lagrangian multipliers in:

$$\min_{\theta} \sum_{j} \mathcal{L}_j(f_\theta) + \lambda \cdot \text{Bias}(f_\theta) + \mu \cdot \text{Toxicity}(f_\theta)[75].$$

Conclusions and Open Challenges

Evaluating and ensuring safe, generalizable behavior in distributed MLLMs remains a multi-faceted challenge. It involves establishing comprehensive benchmarks, adversarial and factual robustness, bias mitigation, and scalable alignment verification [76]. While progress has been made with audit tools and stress benchmarks, open questions persist around long-tail generalization, hallucination control, and global fairness enforcement under federated deployments.

7. Applications, Deployment Architectures, and Real-World Constraints of Distributed MLLMs

While much of the research on multimodal large language models (MLLMs) focuses on training-time innovations and benchmarks, translating these models into production-ready systems introduces a host of deployment challenges [77]. These include distributed inference, latency optimization, edge deployment, energy constraints, and compliance with privacy and fairness regulations [78]. This section outlines practical applications of MLLMs, the architectural patterns used to deploy them, and the systemic constraints shaping their real-world behavior [79].

Applications of Distributed Multimodal LLMs

Distributed MLLMs are increasingly used in industry-scale applications across domains such as:

- **Healthcare:** Multimodal diagnostic models integrating radiology scans, clinical notes, and lab values (e.g., BioGPT-VQA) [80].
- **Autonomous Vehicles:** Fusion of camera, LiDAR, radar, and text maps into real-time policy generation and situational awareness.
- **Retail and E-Commerce:** Vision-language models for product search, personalized recommendations, and cross-modal tagging.
- **Multilingual Education:** Real-time multimodal tutors incorporating diagrams, text, and spoken feedback.
- **Law and Compliance:** Document processing pipelines combining OCR, natural language understanding, and case law retrieval.

For example, in medical imaging, an MLLM might map from $(x_{\text{text}}, x_{\text{image}}) \rightarrow y_{\text{diagnosis}}$ via a distributed ensemble of expert branches trained on partially overlapping data silos across hospitals.

Deployment Architectures: Cloud, Edge, and Hybrid Systems

Depending on latency requirements and privacy constraints, MLLMs are deployed across three primary architectures:

1. **Cloud-Centric Inference:** The model resides entirely in GPU clusters; data is streamed in and results are returned asynchronously.
2. **Edge-Augmented Inference:** Lightweight encoders or modality-specific branches (e.g., image embedding) are run locally, and fused representations are transmitted to a cloud server [81].
3. **Fully On-Device:** In constrained settings (e.g., mobile robotics), quantized and pruned variants of MLLMs are deployed entirely on edge devices.

Let $\mathcal{M} = \{\phi_1, \phi_2, g\}$ denote the encoder-fusion-decoder pipeline. In hybrid systems:

$$\text{Edge: } \phi_i(x_i) \rightarrow \text{Cloud: } g(\phi_1, \ldots, \phi_m) \rightarrow y.$$

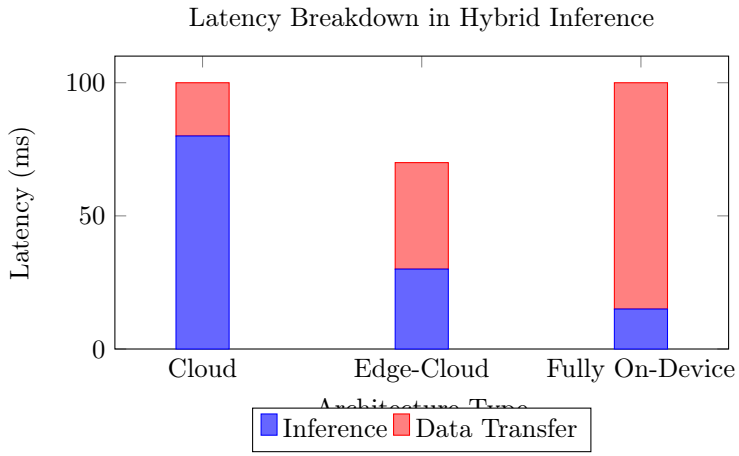

Figure 4: Latency decomposition for different deployment settings [82]. Cloud systems excel in compute speed but suffer from transfer delays; on-device inference minimizes transfer cost but increases compute latency.

Model Optimization for Deployment

To meet deployment constraints, MLLMs must be compressed or adapted without significant performance loss. Standard techniques include:

- **Quantization:** INT8 and FP8 quantization reduce memory and compute demands [83].
- **Distillation:** Teacher-student frameworks that retain cross-modal semantics while reducing model size.
- **Pruning:** Structured pruning of transformer heads, attention blocks, or modality-specific parameters.
- **Mixture-of-Experts (MoE):** Sparse activation routes only a subset of experts during inference.

Let $\mathcal{C}_{\max}$ be the compute budget (e.g., latency or FLOPs). The optimal deployment model f^* solves:

$$f^* = \arg \min_{f \in \mathcal{F}_{\text{compressed}}} \mathcal{L}_{\text{task}}(f) \quad \text{s.t.} \quad \mathcal{C}(f) \leq \mathcal{C}_{\max}.$$

Energy, Privacy, and Regulation-Aware Inference

Energy constraints are paramount in edge applications (e.g., AR glasses or autonomous drones) [84]. Let $\mathcal{E}(f)$ be the energy cost per inference. We define the energy-efficiency metric:

$$\eta(f) = \frac{\mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{M}(f(x), y)]}{\mathcal{E}(f)},$$

which guides deployment selection under power constraints. From a privacy perspective, distributed MLLMs must support:

- **On-device inference without cloud upload.**
- **Differential privacy or secure aggregation when data is shared.**
- **Data anonymization pipelines (e.g., face obfuscation in video).**

Regulatory frameworks such as GDPR or HIPAA impose legal limits on data transfer, retention, and inferencing logic transparency [85, 86]. Compliance-aware deployment includes audit trails, model explainability tools, and selective modality masking:

$$f(x^{(\text{img})}, x^{(\text{text})}) \to f(x^{(\text{img})}, \varnothing) \quad \text{if image is PII-tagged[87].}$$

Real-Time and Streaming Inference Constraints

In streaming applications (e.g., video captioning or live transcription), inference must occur under fixed-time constraints [88]. For input stream $X = \{x_t\}_{t=1}^{T}$, define the real-time constraint:

$$\forall t, \quad \mathbb{E}[\text{Latency}(f(x_t))] < \Delta,$$

where Δ is the maximum tolerable delay (e.g., 100ms for AR). To meet this, MLLMs use techniques such as:

- **Early-exit Transformers** with adaptive halting policies [89].
- **Sliding window or chunked attention** for bounded sequence processing.
- **Lightweight token-level fusion** in lieu of full cross-attention.

Open Deployment Challenges

Despite significant progress, several open issues remain in deploying distributed MLLMs:

- **Heterogeneous Hardware Support:** Efficient MLLM inference across mixed CPU/GPU/TPU/NPU platforms.
- **Bandwidth-Aware Routing:** Jointly optimizing model placement and data routing under variable network conditions [90].
- **Decentralized Monitoring:** Continuous audit and drift detection in federated or offline deployments [91].
- **Human-in-the-Loop Control:** Interface integration for fallback, feedback, and override.

Table 4: Deployment Requirements Across Use Cases

Use Case	**Latency (ms)**	**Energy (mJ)**	**Privacy**	**Real-Time**
Telemedicine Captioning	< 150	Moderate	High	Yes
Drone Navigation	< 50	Low	Medium	Yes
E-commerce Tagging	< 1000	High	Low	No
Document OCR+QA	< 500	Medium	High	No
Smart Glasses Tutor	< 100	Very Low	High	Yes

Summary

Distributed MLLMs present exciting possibilities for cross-modal understanding in real-world applications. However, deployment requires careful trade-offs between latency, memory, privacy, and throughput. Future deployment frameworks will increasingly integrate model compression, streaming attention, privacy-aware routing, and dynamic fusion strategies tailored to the application landscape.

8. Future Directions and Open Research Challenges

As distributed multimodal large language models (MLLMs) continue to evolve, several foundational challenges remain unresolved. These stem from scalability limits, alignment constraints, robustness under modality failure, generalization across domains, and the theoretical understanding of multimodal fusion in distributed environments. In this section, we outline a research agenda spanning algorithmic, systems, and theoretical dimensions, and propose concrete problems and hypotheses to guide future inquiry.

Unified Theoretical Foundations for Multimodal Fusion

Current MLLM designs lack a unified theoretical framework analogous to the universal approximation theorems in unimodal networks [92]. Let $\mathcal{X}_1, \ldots, \mathcal{X}_k$ represent distinct modality spaces and $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathcal{Y}$ a target task. The goal is to characterize the conditions under which a modular architecture:

$$f(x_1, \ldots, x_k) = h\left(\phi_1(x_1), \ldots, \phi_k(x_k)\right),$$

can provably approximate f with bounded generalization error [93]. Open problems include:

- Establishing expressivity bounds for attention-based multimodal fusion layers.
- Characterizing robustness under partially missing or corrupted modalities.

- Developing PAC-style bounds for generalization in distributed, asynchronous MLLMs.

Furthermore, we conjecture that certain fusion strategies (e.g., co-attention) form a strictly more expressive class than others (e.g., simple concatenation) under modality-dependent entropy constraints:

$$\text{If } H(X_i) \ll H(X_j), \text{ then } \text{Attn}(X_i, X_j) > \text{Concat}(X_i, X_j) \text{ in mutual information retention.}$$

Scalable and Communication-Efficient Training Paradigms

Future training paradigms for distributed MLLMs must address high communication cost, memory footprint, and partial supervision [94]. Define $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{task}}$ as the multi-objective loss function. Under federated constraints, only partial gradients $\nabla \mathcal{L}_i$ are available per client. Research directions include:

- Gradient compression for multimodal backpropagation (e.g., quantized tensor aggregation) [95].
- Asynchronous update schemes with delay-compensated fusion ($\theta_{t+1} \leftarrow \theta_t + \eta \sum_i \delta_i^{(t-\tau_i)}$) [96].
- Communication-efficient decentralized distillation (e.g., gossip-based MLLM updates).

Moreover, future systems must leverage modality-specific sparsity for conditional computation. Let $\rho_i(x)$ be the activation routing mask for expert i; minimizing expected communication requires:

$$\min_{\rho} \mathbb{E}_{x \sim \mathcal{D}} \left[\sum_i \rho_i(x) \cdot \text{CommCost}(i) \right] \quad \text{s.t.} \quad \mathcal{L}_{\text{task}}(\rho) \leq \epsilon [97].$$

Multimodal Alignment and Cross-Modal Reasoning

Cross-modal alignment remains a critical bottleneck. In current pipelines, modality pairs (e.g., text-image) are pre-aligned via contrastive loss, but generalizing to higher-order alignment (e.g., video+text+audio+sensor) is poorly

understood. Let $\mathcal{Z}_i$ be latent embeddings for modality i [98]. We define a global alignment condition:

$$\forall (i, j), \quad d(\mathcal{Z}_i, \mathcal{Z}_j) < \delta, \quad \text{where } d(\cdot, \cdot) \text{ is a modality-invariant divergence.}$$

Future models must learn $\mathcal{Z}$ in an unsupervised manner, constrained by both alignment and semantic preservation:

$$\mathcal{L}_{\text{align}} = \sum_{i<j} \text{KL}(\mathcal{Z}_i \parallel \mathcal{Z}_j), \quad \mathcal{L}_{\text{sem}} = \mathbb{E}[\|f(\mathcal{Z}_i) - y\|^2].$$

An important direction is the development of alignment-robust models capable of functioning when modality correspondence is partial, asynchronous, or noisy, a common condition in real-world sensor deployments.

Robustness, Safety, and Interpretability

Distributed MLLMs deployed in the wild are vulnerable to adversarial attacks, modality-specific corruption, and hallucination [99]. Open challenges include:

- Adversarial robustness under cross-modal perturbation (e.g., misleading visual context with consistent textual input) [100].
- Detection and mitigation of hallucinations in open-ended generation (e.g., multimodal grounded factuality scoring).
- Causal interpretability of fusion outcomes (e.g., SHAP or counterfactuals across modality branches) [101].

Let $x = (x_{\text{text}}, x_{\text{img}})$ and $f(x) = y$. A causal counterfactual query is:

$$\text{What would } f \text{ output if } x_{\text{img}} \text{ were replaced with } x'_{\text{img}}?$$

This leads to defining an influence function:

$$I_{\text{img}} = f(x_{\text{text}}, x_{\text{img}}) - f(x_{\text{text}}, x'_{\text{img}}),$$

which quantifies the visual component's salience. Creating interpretable surrogate models for this function is a priority for trustworthy MLLMs [102].

Sustainability and Democratization

As MLLMs grow to trillion-scale parameters, resource disparity becomes a barrier to inclusive research. Future directions must emphasize:

- **Efficient model design:** LoRA-based tuning, zero-shot adapters, and lightweight fusion [103].
- **Open-sourcing decentralized MLLM training frameworks with modular fusion APIs.**
- **Training on decentralized data with verifiable ethical provenance.**
- **Eco-efficient inference:** Real-time energy tracking, dynamic sparsity, and thermal-aware routing.

We hypothesize that the long-term equilibrium of the MLLM ecosystem will shift toward a decentralized, composable architecture of small expert models, coordinated via standard fusion protocols and learned routing controllers:

$$f(x) = g\left(\bigoplus_{i=1}^{k} w_i(x) \cdot \phi_i(x_i)\right), \quad \text{where } w_i(x) \text{ are learned selectors.}$$

Conclusion

Distributed MLLMs are rapidly transforming the landscape of AI research and deployment, yet their full potential remains unrealized [104]. Bridging the gap between theoretical expressiveness, scalable deployment, and safe, interpretable multimodal reasoning constitutes a grand challenge at the intersection of machine learning, distributed systems, and human-centered design. The open problems described herein lay a foundation for future progress and call for a collective effort toward more robust, equitable, and aligned multimodal intelligence.

9. Conclusion

The rapid evolution of large language models and their extension into multimodal domains has fundamentally reshaped the landscape of artificial intelligence research and deployment. This survey has provided a comprehensive overview of distributed large language models (LLMs) and multimodal large language models (MLLMs), elucidating their architectural foundations, training paradigms, and deployment strategies. As the complexity and scale of these models continue to grow, centralized approaches are increasingly constrained by the limitations of computational infrastructure, energy consumption, and data privacy. Distributed approaches—whether through model parallelism, data sharding, federated learning, or expert-specialized architectures—have emerged not merely as engineering solutions, but as essential enablers of scalability, inclusivity, and adaptability in multimodal AI.

We began by formalizing the mathematical principles underlying distributed and multimodal LLMs, highlighting the theoretical gap that remains in fully characterizing their expressive power and convergence behavior under asynchronous, heterogeneous, and partially supervised conditions. Our discussion of architectural trends revealed a proliferation of design choices, including modular encoders, cross-attention fusion layers, sparsely activated expert models, and hybrid hierarchical pipelines that span cloud, edge, and client devices. We surveyed optimization strategies, from traditional gradient-based training to emerging techniques in low-rank adaptation, quantized updates, and decentralized consensus learning—each presenting distinct trade-offs in latency, bandwidth, and convergence stability.

The applications and deployment landscape for distributed MLLMs is vast and rapidly expanding. From real-time multimodal assistants and embodied agents to scientific reasoning, healthcare diagnostics, and autonomous systems, these models are being deployed in increasingly critical settings. As such, the challenges of safety, robustness, interpretability, and ethical alignment have moved to the forefront. Cross-modal hallucination, adversarial vulnerability,

and model collapse under distributional shift are not only technical challenges but also socio-technical risks with real-world consequences.

Looking forward, the path to building general-purpose, trustworthy, and resource-efficient distributed MLLMs requires bridging several open research gaps. These include developing unified theoretical frameworks for multimodal fusion, scalable training algorithms with bounded communication overhead, dynamic routing mechanisms for conditional computation, and interpretable reasoning pathways across modalities. Additionally, there is an urgent need to democratize access to MLLM development by lowering the computational barrier to entry, incentivizing open-source collaboration, and formalizing standards for fairness, safety, and environmental sustainability.

In conclusion, distributed multimodal LLMs represent one of the most promising frontiers in AI, blending advances in machine learning, systems engineering, cognitive science, and ethics. This survey provides a foundational reference for researchers and practitioners aiming to contribute to this multifaceted domain. By fostering collaboration across disciplines and investing in principled innovation, we can realize a future where large-scale multimodal intelligence is not only powerful but also responsible, inclusive, and aligned with human values.

References

[1] Z. Li, H. Zhu, Z. Lu, M. Yin, Synthetic data generation with large language models for text classification: Potential and limitations, arXiv preprint arXiv:2310.07849 (2023).

[2] Z. Shuai, L. Shen, Mitigating heterogeneity in federated multimodal learning with biomedical vision-language pre-training, arXiv preprint arXiv:2404.03854 (2024).

[3] A. Seth, M. Hemani, C. Agarwal, Dear: Debiasing vision-language models with additive residuals, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6820–6829.

[4] Z. Zhang, D. Cai, Y. Zhang, M. Xu, S. Wang, A. Zhou, Fedrdma: Communication-efficient cross-silo federated LLM via chunked rdma transmission, in: Proceedings of the 4th Workshop on Machine Learning and Systems, 2024, pp. 126–133.

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (8) (2019) 9.

[6] T. Shen, Z. Li, Z. Zhao, D. Zhu, Z. Lv, S. Zhang, K. Kuang, F. Wu, An adaptive aggregation method for federated learning via meta controller, in: Proceedings of the 6th ACM International Conference on Multimedia in Asia Workshops, 2024, pp. 1–1.

[7] S. Yu, J. P. Muñoz, A. Jannesari, Federated foundation models: Privacy-preserving and collaborative learning for large models, arXiv preprint arXiv:2305.11414 (2023).

[8] J. He, P. Li, G. Liu, S. Zhong, Parameter-efficient fine-tuning medical multimodal large language models for medical visual grounding, arXiv preprint arXiv:2410.23822 (2024).

[9] A. Dosovitskiy, An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

[10] R. Kazmierczak, E. Berthier, G. Frehse, G. Franchi, Explainability for vision foundation models: A survey, Available at SSRN 5106267 (2025).

[11] J. Zhao, Privacy-preserving fine-tuning of artificial intelligence (ai) foundation models with federated learning, differential privacy, offsite tuning, and parameter-efficient fine-tuning (peft), Authorea Preprints (2023).

[12] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, Exploiting linear structure within convolutional networks for efficient evaluation, Advances in neural information processing systems 27 (2014).

[13] A. Lou, C. Meng, S. Ermon, Discrete diffusion modeling by estimating the ratios of the data distribution, arXiv preprint arXiv:2310.16834 (2023).

[14] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).

[15] S. Wang, T. Xu, H. Li, C. Zhang, J. Liang, J. Tang, P. S. Yu, Q. Wen, Large language models for education: A survey and outlook, arXiv preprint arXiv:2403.18105 (2024).

[16] F. Zeng, W. Gan, Y. Wang, S. Y. Philip, Distributed training of large language models, in: 2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2023, pp. 840–847.

[17] L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu, et al., A survey on large language models for recommendation, World Wide Web 27 (5) (2024) 60.

[18] Y. Yao, T. Yu, A. Zhang, C. Wang, J. Cui, H. Zhu, T. Cai, H. Li, W. Zhao, Z. He, et al., Minicpm-v: A gpt-4v level mllm on your phone, arXiv preprint arXiv:2408.01800 (2024).

[19] Y. Yuan, Z. Li, B. Zhao, A survey of multimodal learning: Methods, applications, and future, ACM Computing Surveys (2025).

[20] H. Liu, et al., Visual instruction tuning, Advances in Neural Information Processing Systems (NeurIPS) 36 (2023).

[21] C. Schlarmann, N. D. Singh, F. Croce, M. Hein, Robust clip: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models, in: International Conference on Machine Learning, PMLR, 2024, pp. 43685–43704.

[22] O. Gafni, A. Polyak, O. Ashual, E. Shechtman, T. Park, Make-a-scene: Scene-based text-to-image generation with human priors, in: European Conference on Computer Vision (ECCV), 2022, pp. 39–55.

[23] Y. Wang, Y. Lin, X. Zeng, G. Zhang, Privatelora for efficient privacy preserving LLM, arXiv preprint arXiv:2311.14030 (2023).

[24] H. Woisetschläger, A. Erben, S. Wang, R. Mayer, H.-A. Jacobsen, Federated fine-tuning of LLMs on the very edge: The good, the bad, the ugly, in: Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning, 2024, pp. 39–50.

[25] X. Guo, Y. Chen, Generative ai for synthetic data generation: Methods, challenges and the future, arXiv preprint arXiv:2403.04190 (2024).

[26] J. Li, J. Xu, S. Huang, Y. Chen, W. Li, J. Liu, Y. Lian, J. Pan, L. Ding, H. Zhou, et al., Large language model inference acceleration: A comprehensive hardware perspective, arXiv preprint arXiv:2410.04466 (2024).

[27] H. Zeng, Z. Yue, Y. Zhang, L. Shang, D. Wang, Fair federated learning with biased vision-language models, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 10002–10017.

[28] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, X. Wang, Groupvit: Semantic segmentation emerges from text supervision, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 18134–18144.

[29] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, J. Gao, Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts, arXiv preprint arXiv:2310.02255 (2023).

[30] Z. Qin, Z. Wu, B. He, S. Deng, Federated data-efficient instruction tuning for large language models, arXiv preprint arXiv:2410.10926 (2024).

[31] M. Javaheripi, S. Bubeck, M. Abdin, J. Aneja, S. Bubeck, C. C. T. Mendes, W. Chen, A. Del Giorno, R. Eldan, S. Gopi, et al., Phi-2: The surprising power of small language models, Microsoft Research Blog 1 (3) (2023) 3.

[32] Z. Lin, X. Hu, Y. Zhang, Z. Chen, Z. Fang, X. Chen, A. Li, P. Vepakomma, Y. Gao, Splitlora: A split parameter-efficient fine-tuning framework for large language models, arXiv preprint arXiv:2407.00952 (2024).

[33] L. Sani, A. Iacob, Z. Cao, B. Marino, Y. Gao, T. Paulik, W. Zhao, W. F. Shen, P. Aleksandrov, X. Qiu, et al., The future of large language model pre-training is federated, arXiv preprint arXiv:2405.10853 (2024).

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

[35] T. Fan, Y. Kang, G. Ma, W. Chen, W. Wei, L. Fan, Q. Yang, Fate-llm: A industrial grade federated learning framework for large language models, arXiv preprint arXiv:2310.10049 (2023).

[36] R. Ye, W. Wang, J. Chai, D. Li, Z. Li, Y. Xu, Y. Du, Y. Wang, S. Chen, Openfedllm: Training large language models on decentralized private data via federated learning, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6137–6147.

[37] S. K. Pentyala, Z. Wang, B. Bi, K. Ramnath, X.-B. Mao, R. Radhakrishnan, S. Asur, et al., Paft: A parallel training paradigm for effective LLM fine-tuning, arXiv preprint arXiv:2406.17923 (2024).

[38] J. Hagemann, S. Weinbach, K. Dobler, M. Schall, G. de Melo, Efficient parallelization layouts for large-scale distributed model training, arXiv preprint arXiv:2311.05610 (2023).

[39] J. Koo, M. Jang, J. Ok, Towards robust and efficient federated low-rank adaptation with heterogeneous clients, arXiv preprint arXiv:2410.22815 (2024).

[40] L. Abrahamyan, Y. Chen, G. Bekoulis, N. Deligiannis, Learned gradient compression for distributed deep learning, IEEE Transactions on Neural Networks and Learning Systems 33 (12) (2021) 7330–7344.

[41] K. Zhou, J. Yang, C. C. Loy, Z. Liu, Conditional prompt learning for vision-language models, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16816–16825.

[42] J. Xu, Z. Li, W. Chen, Q. Wang, X. Gao, Q. Cai, Z. Ling, On-device language models: A comprehensive review, arXiv preprint arXiv:2409.00088 (2024).

[43] M. Z. Hossain, A. Imteaj, Securing vision-language models with a robust encoder against jailbreak and adversarial attacks, in: 2024 IEEE International Conference on Big Data (BigData), IEEE, 2024, pp. 6250–6259.

[44] M. Fahad, M. Shojafar, M. Abbas, I. Ahmed, H. Ijaz, A multi-queue priority-based task scheduling algorithm in fog computing environment, Concurrency and Computation: Practice and Experience 34 (28) (2022) e7376.

[45] A. Nabli, L. Fournier, P. Erbacher, L. Serrano, E. Belilovsky, E. Oyallon, Acco: Accumulate while you communicate, hiding communications in distributed LLM training, arXiv preprint arXiv:2406.02613 (2024).

[46] F. Brakel, U. Odyurt, A.-L. Varbanescu, Model parallelism on distributed infrastructure: A literature review from theory to LLM case-studies, arXiv preprint arXiv:2403.03699 (2024).

[47] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li, et al., Regionclip: Region-based language-image pretrain-

ing, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16793–16803.

[48] K. Yan, Z. Wei, L. Xinyu, Grounding foundation models through federated transfer learning: A general framework, arXiv preprint arXiv:2311.17431 (2023).

[49] Z. Qin, D. Chen, B. Qian, B. Ding, Y. Li, S. Deng, Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes, arXiv preprint arXiv:2312.06353 (2023).

[50] Y. Bai, Y. Zhang, J. Yang, J. Liu, J. Tang, J. Wu, J. Gao, J. Wang, Binarybert: Pushing the limit of bert quantization, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 4334–4343.

[51] Y. Wang, J. You, Y. Li, Dynamic resource aggregation method based on statistical capacity distribution, Electronics 13 (23) (2024) 4617.

[52] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).

[53] X. Ren, J. Tang, D. Yin, N. Chawla, C. Huang, A survey of large language models for graphs, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6616–6626.

[54] P. Yanghe, C. Jun, D. Linjun, Z. Xiaobo, Z. Hongyan, Cloud-edge collaborative large model services: Challenges and solutions, IEEE Network (2024).

[55] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large language models: A survey, arXiv preprint arXiv:2402.06196 (2024).

[56] Z. Wang, Z. Shen, Y. He, G. Sun, H. Wang, L. Lyu, A. Li, Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations, arXiv preprint arXiv:2409.05976 (2024).

[57] Y. Zniyed, T. P. Nguyen, et al., Efficient tensor decomposition-based filter pruning, Neural Networks 178 (2024) 106393.

[58] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, I. Stoica, Spark: Cluster computing with working sets, in: 2nd USENIX workshop on hot topics in cloud computing (HotCloud 10), 2010.

[59] Z. Li, W. Feng, M. Guizani, H. Yu, Tpi-llm: Serving 70b-scale LLMs efficiently on low-resource edge devices, arXiv preprint arXiv:2410.00531 (2024).

[60] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, K.-W. Chang, Visualbert: A simple and performant baseline for vision and language, arXiv preprint arXiv:1908.03557 (2019).

[61] Y. He, X. Zhang, J. Sun, Channel pruning for accelerating very deep neural networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 1389–1397.

[62] J. Chen, W. Li, G. Yang, X. Qiu, S. Guo, Federated learning meets edge computing: A hierarchical aggregation mechanism for mobile devices, in: International Conference on Wireless Algorithms, Systems, and Applications, Springer, 2022, pp. 456–467.

[63] S. Wang, B. Z. Li, M. Khabsa, H. Fang, H. Ma, Linformer: Self-attention with linear complexity, arXiv preprint arXiv:2006.04768 (2020).

[64] A. Imteaj, M. Z. Hossain, S. Zaman, A. R. Shahid, Tripleplay: Enhancing federated learning with clip for non-iid data and resource efficiency, 2024 IEEE Conference on Big Data (2024).

[65] J. Kuang, Y. Shen, J. Xie, H. Luo, Z. Xu, R. Li, Y. Li, X. Cheng, X. Lin, Y. Han, Natural language understanding and inference with MLLM in visual question answering: A survey, ACM Computing Surveys (2025).

[66] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks (2018). arXiv:1812.06127.

[67] N. Kitaev, Ł. Kaiser, A. Levskaya, Reformer: The efficient transformer, arXiv preprint arXiv:2001.04451 (2020).

[68] A. Raje, Communication-efficient LLM training for federated learning, Ph.D. thesis, Ph. D. thesis, Carnegie Mellon University Pittsburgh, PA (2024).

[69] Y. Zhou, M. Fritz, M. Keuper, Multimax: Sparse and multi-modal attention learning, arXiv preprint arXiv:2406.01189 (2024).

[70] K. He, R. Mao, Q. Lin, Y. Ruan, X. Lan, M. Feng, E. Cambria, A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics, arXiv preprint arXiv:2310.05694 (2023).

[71] Y. Zhao, D. Wu, J. Wang, Alisa: Accelerating large language model inference via sparsity-aware kv caching, arXiv preprint arXiv:2403.17312 (2024).

[72] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, et al., Pythia: A suite for analyzing large language models across training and scaling, in: International Conference on Machine Learning, PMLR, 2023, pp. 2397–2430.

[73] M. Huang, A. Shen, K. Li, H. Peng, B. Li, H. Yu, Edgellm: A highly efficient cpu-fpga heterogeneous edge accelerator for large language models, arXiv preprint arXiv:2407.21325 (2024).

[74] S. Kombrink, T. Mikolov, M. Karafiát, L. Burget, Recurrent neural network based language modeling in meeting recognition., in: Interspeech, Vol. 11, 2011, pp. 2877–2880.

[75] B. Hu, J. Li, L. Xu, M. Lee, A. Jajoo, G.-W. Kim, H. Xu, A. Akella, Blockllm: Multi-tenant finer-grained serving for large language models, arXiv preprint arXiv:2404.18322 (2024).

[76] J. Xie, Y. Zhang, M. Lin, L. Cao, R. Ji, Advancing multimodal large language models with quantization-aware scale learning for efficient adaptation, in: Proceedings of the 32nd ACM International Conference on Multimedia, 2024, pp. 10582–10591.

[77] P. Ganesh, D. Ramatlongo, N. Pourdamghani, W. Guo, N. Constant, S. Parthasarathy, H. Sajjad, G. Riccardi, M. Faruqui, Compressing large-scale transformer-based models: A case study on bert, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2021, pp. 5290–5306.

[78] L. Yao, J. Han, Y. Wen, X. Liang, D. Xu, W. Zhang, Z. Li, C. Xu, H. Xu, Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection, Advances in Neural Information Processing Systems 35 (2022) 9125–9138.

[79] H. Zhang, S. Li, F. Ye, M. Fang, J. Zhao, Y.-H. Chan, E. C.-H. Ngai, T. Voigt, Distributed foundation models for multi-modal learning in 6g wireless networks, IEEE Communications Magazine 62 (6) (2024) 20–26.

[80] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, D. Zhou, Mobilebert: a compact task-agnostic bert for resource-limited devices, arXiv preprint arXiv:2004.02984 (2020).

[81] Y. Chen, T. Zhang, X. Jiang, Q. Chen, C. Gao, W. Huang, Fedbone: Towards large-scale federated multi-task learning, arXiv preprint arXiv:2306.17465 (2023).

[82] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, Y. Zhang, A survey on large language model (llm) security and privacy: The good, the bad, and the ugly, High-Confidence Computing (2024) 100211.

[83] C. Guo, F. Cheng, Z. Du, J. Kiessling, J. Ku, S. Li, Z. Li, M. Ma, T. Molom-Ochir, B. Morris, et al., A survey: Collaborative hardware and software design in the era of large language models, arXiv preprint arXiv:2410.07265 (2024).

[84] X. Shen, Z. Kong, C. Yang, Z. Han, L. Lu, P. Dong, C. Lyu, C.-h. Li, X. Guo, Z. Shu, et al., Edgeqat: Entropy and distribution guided quantization-aware training for the acceleration of lightweight LLMs on the edge, arXiv preprint arXiv:2402.10787 (2024).

[85] C. Lyu, M. Wu, L. Wang, X. Huang, B. Liu, Z. Du, S. Shi, Z. Tu, Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration, arXiv preprint arXiv:2306.09093 (2023).

[86] Y. Zniyed, T. P. Nguyen, et al., Enhanced network compression through tensor decompositions and pruning, IEEE Transactions on Neural Networks and Learning Systems (2024).

[87] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé, et al., Gemma 2: Improving open language models at a practical size, arXiv preprint arXiv:2408.00118 (2024).

[88] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, Y. Zhao, Resource-efficient federated learning with hierarchical aggregation in edge computing, in: IEEE INFOCOM 2021-IEEE conference on computer communications, IEEE, 2021, pp. 1–10.

[89] N. Dey, G. Gosal, H. Khachane, W. Marshall, R. Pathria, M. Tom, J. Hestness, et al., Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster, arXiv preprint arXiv:2304.03208 (2023).

[90] Z. Lan, Albert: A lite bert for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942 (2019).

[91] D. P. Nguyen, J. P. Munoz, A. Jannesari, Flora: Enhancing vision-language models with parameter-efficient federated learning, arXiv preprint arXiv:2404.15182 (2024).

[92] J. Rasley, S. Rajbhandari, O. Ruwase, S. Yang, Y. Zhang, Y. He, Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters, arXiv preprint arXiv:2007.00072 (2020).

[93] J. Zheng, H. Zhang, L. Wang, W. Qiu, H. Zheng, Z. Zheng, Safely learning with private data: A federated learning framework for large language model, arXiv preprint arXiv:2406.14898 (2024).

[94] A. Khoshsirat, G. Perin, M. Rossi, Decentralized LLM inference over edge networks with energy harvesting, arXiv preprint arXiv:2408.15907 (2024).

[95] R. J. Das, M. Sun, L. Ma, Z. Shen, Beyond size: How gradients shape pruning decisions in large language models, arXiv preprint arXiv:2311.04902 (2023).

[96] P. Petoumenos, L. Mukhanov, Z. Wang, H. Leather, D. S. Nikolopoulos, Power capping: What works, what does not, in: 2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2015, pp. 525–534.

[97] T. Guo, S. Guo, J. Wang, Pfedprompt: Learning personalized prompt for vision-language models in federated learning, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 1364–1374.

[98] R. Gao, T.-H. Oh, K. Grauman, L. Torresani, Listen to look: Action recognition by previewing audio, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10457–10467.

[99] N. Mu, A. Kirillov, D. Wagner, S. Xie, Slip: Self-supervision meets language-image pre-training, in: European conference on computer vision, Springer, 2022, pp. 529–544.

[100] A. Gu, T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, arXiv preprint arXiv:2312.00752 (2023).

[101] H. Laurençon, L. Tronchon, M. Cord, V. Sanh, What matters when building vision-language models?, Advances in Neural Information Processing Systems 37 (2024) 87874–87907.

[102] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, et al., Rethinking attention with performers, arXiv preprint arXiv:2009.14794 (2020).

[103] OpenAI, Gpt-4v system card, accessed: 2024-10-29 (2023).
URL `https://openai.com/index/gpt-4v-system-card/`

[104] D. Chen, D. Gao, W. Kuang, Y. Li, B. Ding, PFL-bench: A comprehensive benchmark for personalized federated learning, Advances in Neural Information Processing Systems 35 (2022) 9344–9360.