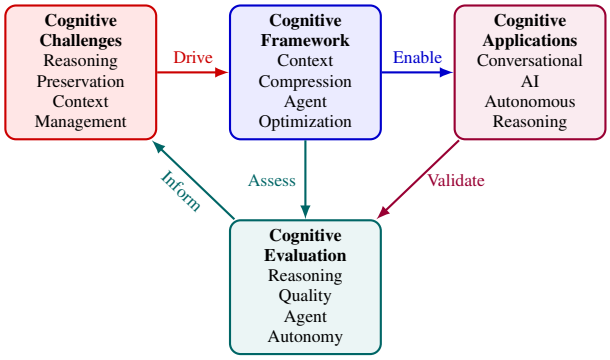
Cognitive Edge Computing: A Comprehensive Survey on Optimizing Large Models and AI Agents for Pervasive Deployment

Xubin Wang, Qing Li, *Fellow, IEEE* and Weijia Jia*, *Fellow, IEEE*

***Abstract*—This article surveys Cognitive Edge Computing as a practical and methodical pathway for deploying reasoning-capable Large Language Models (LLMs) and autonomous AI agents on resource-constrained devices at the network edge. We present a unified, cognition-preserving framework spanning: (1) model optimization (quantization, sparsity, low-rank adaptation, distillation) aimed at retaining multi-step reasoning under tight memory/compute budgets; (2) system architecture (on-device inference, elastic offloading, cloud–edge collaboration) that trades off latency, energy, privacy, and capacity; and (3) adaptive intelligence (context compression, dynamic routing, federated personalization) that tailors computation to task difficulty and device constraints. We synthesize advances in efficient Transformer design, multimodal integration, hardware-aware compilation, privacy-preserving learning, and agentic tool use, and map them to edge-specific operating envelopes. We further outline a standardized evaluation protocol covering latency, throughput, energy per token, accuracy, robustness, privacy, and sustainability, with explicit measurement assumptions to enhance comparability. Remaining challenges include modality-aware reasoning benchmarks, transparent and reproducible energy reporting, edge-oriented safety/alignment evaluation, and multi-agent testbeds. We conclude with practitioner guidelines for cross-layer co-design of algorithms, runtime, and hardware to deliver reliable, efficient, and privacy-preserving cognitive capabilities on edge devices.**

***Index Terms*—Edge Computing, Large Language Models, Small Language Models, Quantization, Knowledge Distillation, Cloud–Edge Collaboration**



Cognitive Edge Computing: Closed-loop optimization for reasoning-preserving deployment

Fig. 1. **Cognitive Edge Computing Framework:** Integrated approach for deploying reasoning-capable LLMs and autonomous Agents on resource-constrained edge devices. The framework consists of four interconnected components: (1) *Cognitive Challenges* (red) addressing reasoning preservation and context management under resource constraints; (2) *Cognitive Framework* (blue) implementing context compression and agent optimization techniques; (3) *Cognitive Applications* (purple) enabling conversational AI and autonomous reasoning capabilities; and (4) *Cognitive Evaluation* (teal) assessing reasoning quality and agent autonomy. The closed-loop design ensures continuous improvement through feedback mechanisms where evaluation results inform challenge identification and application validation drives framework refinement.

I. Introduction

The convergence of LLMs and AI agents with edge computing heralds the emergence of *Cognitive Edge Computing*—a revolutionary paradigm that brings sophisticated cognitive capabilities directly to resource-constrained devices at the network periphery [1], [2]. Unlike traditional edge computing that focuses primarily on data processing and basic analytics, *Cognitive Edge Computing* represents a fundamental shift toward deploying advanced AI systems that can understand context, reason autonomously, and make intelligent decisions in real-time, all while operating within the severe constraints of edge environments [3], [4].

We define *Cognitive Edge Computing* as the intelligent orchestration of advanced AI models and autonomous agents across heterogeneous computing hierarchies, enabling cognitive tasks such as natural language understanding, multimodal reasoning, and adaptive decision-making at the network edge. This paradigm transcends conventional edge AI by emphasizing not just computational efficiency, but the preservation of sophisticated cognitive functions including contextual awareness, reasoning quality, and autonomous behavior [5].

To clarify the distinction, traditional edge AI primarily focuses on narrow perception tasks such as image classification, object detection, or basic analytics, which require relatively simple computational models and can operate with limited cognitive capabilities [6]. In contrast, cognitive edge computing targets open-domain, often multi-modal reasoning tasks that demand advanced cognitive functions like multi-step reasoning, contextual understanding, and autonomous decision-making, all while maintaining human-level performance under stringent resource constraints.

The transformative potential of *Cognitive Edge Computing* stems from recent breakthroughs in foundation models and autonomous systems. LLMs have evolved from task-specific pipelines to general architectures capable of in-context learning, multi-step reasoning, and autonomous goal pursuit [7]–[9]. Breakthrough models such as GPT-3 (175B parameters) [7], PaLM (540B parameters) [10], and open foundation models including Deepseek-r1 [11] have demonstrated unprecedented capabilities in natural language understanding, code generation,

Corresponding authors (*): Weijia Jia.

Xubin Wang and Weijia Jia are with the Beijing Normal-Hong Kong Baptist University and the Institute of Artificial Intelligence and Future Networks, Beijing Normal University (Zhuhai campus). E-mail: wangxubin@ieee.org (Xubin Wang); jiawj@bnu.edu.cn (Weijia Jia).

Qing Li is currently a Chair Professor (Data Science) and the Head of the Department of Computing, the Hong Kong Polytechnic University. E-mail: qing-prof.li@polyu.edu.hk

Manuscript received September 18, 2025; revised September 18, 2025.

mathematical reasoning, and complex problem-solving. Concurrently, autonomous agent research has advanced sophisticated frameworks for goal-oriented behavior, adaptive planning, tool orchestration, and multi-agent coordination [12]–[14].

However, deploying these advanced AI systems at the edge presents unprecedented challenges that traditional optimization techniques cannot adequately address. *Cognitive Edge Computing* requires a fundamental rethinking of how we approach AI deployment, moving beyond simple model compression to comprehensive frameworks that preserve cognitive capabilities while ensuring real-time performance, energy efficiency, and privacy protection.

Recent works advance complementary building blocks spanning on-device personalization and data selection [15], compression and inference under tight memory/compute constraints [16]–[21], collaborative serving and routing across edge–cloud and MoE systems [22]–[25], edge hardware acceleration and memory systems (FPGA/NPU/PIM/flash-assisted) [26]–[34], domain security and compliance (Internet of Things (IoT) fuzzing and medical regulation) [35]–[37], and application exemplars from multimodal edge LLMs to private IR and industrial IoT [38]–[41]. We integrate these strands within a unified, reasoning-preserving framework for cognitive edge computing.

Market Dynamics and On-Device AI Evolution: The global edge AI market is projected to grow rapidly, driven by demand in manufacturing, automotive, consumer electronics, and healthcare. Edge-side intelligence offers low latency, offline availability, energy efficiency, enhanced privacy, and personalized experiences [42]. Since 2023, sub-10B parameter models like Meta's LLaMA, Microsoft's Phi, Google's Gemma, and Nexa AI's Octopus have accelerated on-device deployment, leveraging MoE routing, quantization, and compression for mobile constraints [43]–[47].

Computing Architecture Hierarchy: We define three distinct tiers in the computing architecture hierarchy to contextualize the deployment of cognitive edge computing systems: *Cloud* (remote data centers with abundant computational resources, enabling unlimited scalability and complex processing); *Edge* (servers and base stations with tens to hundreds of GB memory, providing intermediate processing capabilities closer to data sources); *Device/Client-side* (end-user devices with GB-scale memory and W-scale power budgets, supporting direct local inference). "On-Device Large Language Models" specifically refer to models deployed directly on client terminals, enabling offline operation with potential cloud collaboration for enhanced capabilities [3], [48], [49].

Definition and Taxonomy of Edge-Side Large Models: Edge-side large models, also known as on-device LLMs, are pre-trained Transformer-based architectures optimized for deployment on resource-constrained edge devices through compression techniques such as quantization, pruning, knowledge distillation, and low-rank approximation [50]. These techniques significantly reduce the computational and memory footprint compared to cloud-scale models, enabling efficient local inference [51]. Representative implementations include specialized architectures[1], mixture-of-experts models[2], and ultra-lightweight designs[3], demonstrating diverse approaches to edge deployment optimization.

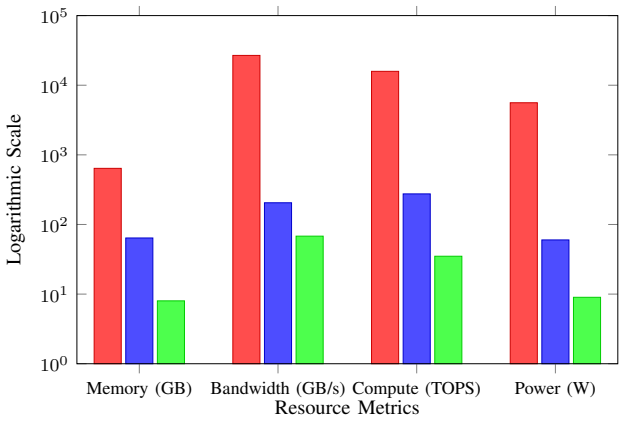


Fig. 2. Illustrative resource comparison (cloud accelerator cluster vs. edge server vs. mobile SoC). Values approximate public peak specs; actual usable throughput depends on workload, precision, and batching [57]–[59].

The convergence of these advanced AI capabilities with edge infrastructure enables the deployment of general-purpose reasoning, language understanding, and autonomous decision-making directly at the network periphery [1], [52], [53]. Cognitive edge computing must operate under stringent energy, memory, and latency constraints while maintaining human-level cognitive performance [4], [54]. Figure 1 illustrates the closed-loop interaction among the fundamental challenges, optimization strategies, target applications, and evaluation frameworks that define this emerging field.

A. Survey Methodology and Evidence Grading

We surveyed peer-reviewed venues, benchmarks, and high-impact preprints (2018–2025, emphasis 2023–2025) using keywords like "edge LLM", "quantization", "knowledge distillation", etc. Inclusion prioritized works with methodological details or artifacts; exclusion for unverifiable sources. Evidence graded as E1 (archival with replication), E2 (peer-reviewed with partial artifacts), E3 (industry/preprint).

At the core of cognitive edge computing lies a fundamental deployment contradiction: LLM and agent requirements exceed edge capabilities by orders of magnitude [7], [55], [56]. Figure 2 illustrates this resource disparity across computing tiers.

Addressing this contradiction requires coordinated optimization across model, system, and collaboration layers rather than isolated compression. Figure 1 presents our cognitive edge computing framework, while Figure 3 structures the solution space into: data optimization (cleaning, augmentation, bias mitigation), model optimization (quantization, sparsity/pruning, distillation, low-rank + architecture tailoring, emergence of Small Language Models (SLMs)), and system/runtime optimization (partitioning, scheduling, collaborative / hybrid

[1]MobileLLM-R1: https://huggingface.co/collections/facebook/mobilellm-r1-68c4597b104fac45f28f448e
[2]Ring-mini-2.0: https://huggingface.co/inclusionAI/Ring-mini-2.0
[3]Tiny-random-Llama: https://huggingface.co/HuggingFaceH4/tiny-random-LlamaForCausalLM

routing, federated adaptation). These layers interact: early wins (e.g., 4–8 bit quantization) ease bandwidth pressure; additional sparsity then shows diminishing returns unless placement and scheduling co-adapt. Collaboration patterns (large–small cascades, confidence gating, co-evolution) further trade accuracy, latency, and energy.

Edge scenarios with privacy-sensitive, latency-critical cognition—vehicular navigation, clinical triage, industrial diagnostics, biomarker-driven screening, civic infrastructure—drive locality for personalization and resilience [5], [6], [60], [61]. Direct on-device LLM execution reduces interactive latency and raw data exposure [62], [63]; hybrid designs still offload heavy context expansion or long-horizon reasoning when beneficial [2]. Emerging hardware (specialized NPUs, low-latency fabrics, neuromorphic exploration) enables but does not replace principled co-design.

Despite rapid progress, prior surveys largely cover traditional edge ML or generic LLM optimization in cloud contexts, leaving a gap in synthesizing cross-layer techniques explicitly targeting reasoning preservation under edge constraints [5], [6], [43]. This survey fills that gap by unifying cognitive workload characteristics, compression + architecture tailoring methods, system/runtime orchestration, and large–small cooperation strategies within a single evaluative framework.

Industry and ecosystem snapshot: Recent industry analyses indicate multi-factor tailwinds for on-device AI: national and municipal policies emphasizing intelligent terminals, rapid hardware advances (e.g., 40+ TOPS NPUs for AI PCs; LPDDR5X/5T with 10.7 GT/s), and flagship product cycles (Apple Intelligence in iPhone, HarmonyOS with Ascend/Atlas, Snapdragon X/8 Gen platforms). These forces collectively accelerate deployment across AI PCs, smartphones, wearables, smart homes, automotive, and industrial equipment, with edge–cloud collaboration as the default operating model for balancing capability, latency, and privacy [4].

B. Scope and Summary of Surveyed Contributions

We organize existing literature rather than claim new algorithms. Our contributions:

- **Problem framing:** Clarifies cognitive edge objectives (latency, energy, reasoning fidelity, privacy) distinct from conventional accuracy-only targets.
- **Technique taxonomy:** Integrates compression (quantization, sparsity, distillation, low-rank), architectural tailoring (SLMs, efficient attention), and system orchestration (partitioning, routing, federated adaptation).
- **Cross-layer view:** Highlights interaction and diminishing returns across stacked optimizations.
- **Large–small collaboration patterns:** Summarizes routing / co-evolution designs with reported (heterogeneous) performance ranges.
- **Security and trust lens:** Aggregates attack vectors and mitigation overheads relevant to constrained deployments.
- **Gap analysis:** Identifies needs for reproducible energy reporting, standardized cognitive benchmarks, resource-aware XAI, and multi-agent edge testbeds.

Table I summarizes key deployment characteristics across computing tiers based on published benchmarks and hardware specifications. Throughout this survey, we prefer standard terminology (e.g., "on-device LLM", "energy per request") over neologisms with "Cognitive" prefixes for metrics/categories.

II. Foundational Concepts for Cognitive Edge Computing

This section examines the foundational concepts underpinning cognitive edge computing: the evolution of edge AI, LLM characteristics, and AI agent properties. Figure 4 summarizes the end-to-end pipeline from multi-modal inputs to agent actions.

A. On-Device LLMs Evolution Timeline

The trajectory of on-device LLMs represents a fundamental paradigm shift from cloud-dependent to autonomous edge AI capabilities [43]. This evolution demonstrates how architectural innovations, compression techniques, and hardware optimization converge to enable sophisticated language understanding directly on resource-constrained devices.

2023: The Foundation Year The year 2023 marked the beginning of practical on-device LLM deployment with the emergence of sub-10B parameter models. Meta's LLaMA series [44] pioneered efficient transformer architectures through innovations like RMSNorm and grouped-query attention (GQA), optimizing for reduced computational and memory requirements while maintaining competitive performance. Microsoft's Phi series (e.g., Phi-1 at 1.3B parameters) [45] demonstrated that carefully curated, high-quality training data could achieve remarkable capabilities despite a highly compact model size. This period also saw the rise of other compact models like ChatGLM [64] and Qwen [65], establishing the core principles of efficiency-oriented design.

2024: Acceleration and Diversification The on-device LLM landscape expanded dramatically in 2024 with specialized model families addressing diverse deployment scenarios. Google's Gemini Nano [46] integrated multimodal capabilities within mobile-optimized architectures, enabling real-time image-text understanding on smartphones. The field saw intense innovation in efficiency, with models like Nexa AI's Octopus series [47] reporting breakthroughs in function calling efficiency, and Apple's OpenELM [66] showcasing a layer-wise scaling strategy. The development of highly efficient inference engines like `LLMCad` [67] and the architectural refinements proposed in `MobileLLM` [68] further pushed the boundaries of what was possible on-device. Performance evaluations on commercial smartphones [69] began to provide crucial empirical data for the field.

2025: Maturation and Widespread Integration By 2025, the on-device LLM ecosystem reached a critical maturation point with widespread commercial adoption. Major technology companies integrated sub-10B parameter models into flagship products, achieving real-time conversational AI with sub-500ms

[4]Dongxing Technology Research Report: https://pdf.dfcfw.com/pdf/H3_AP202409271640082519_1.pdf

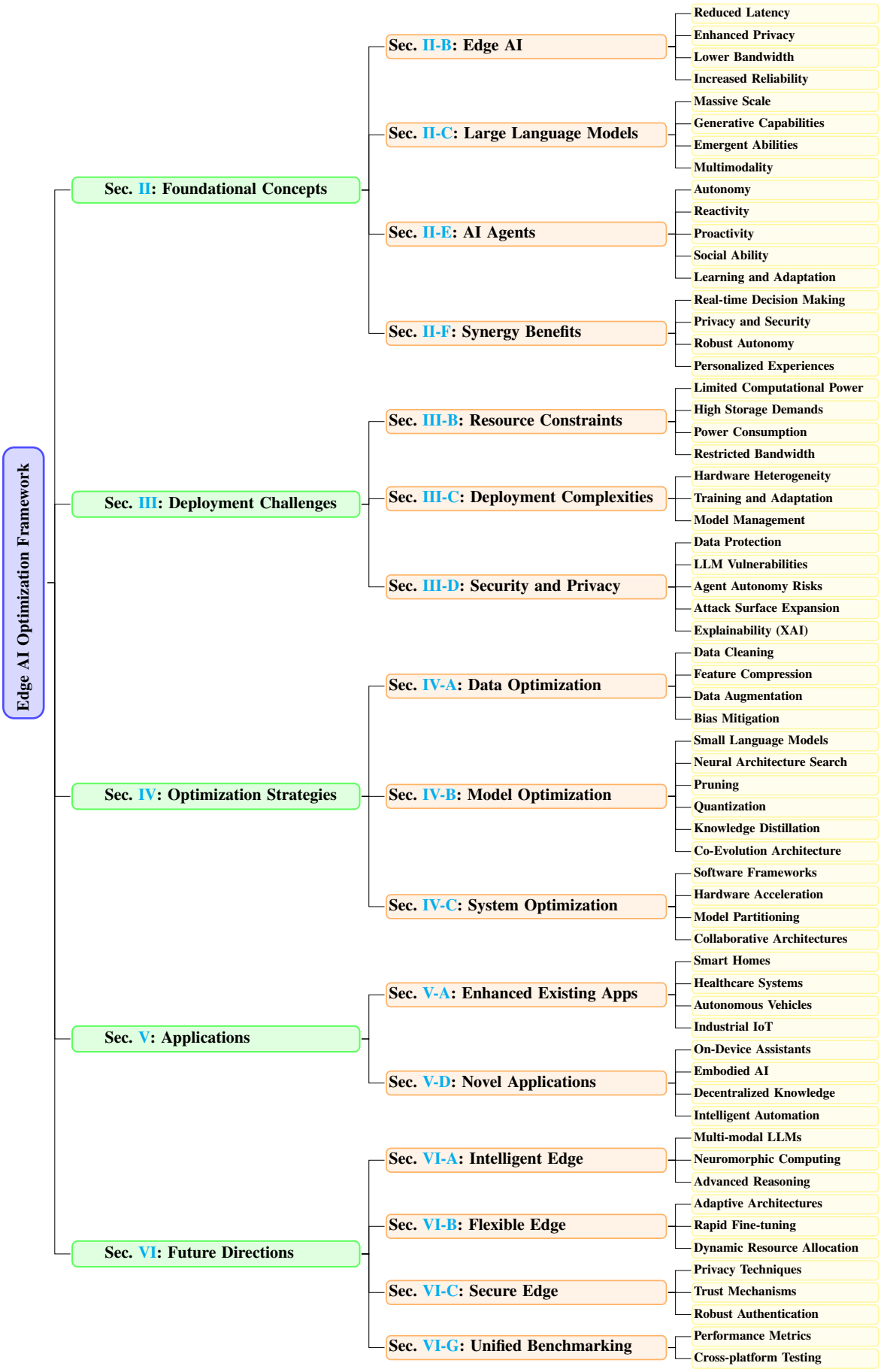


Fig. 3. Cognitive Edge AI Optimization Framework for LLMs and Agents

TABLE I
DEPLOYMENT CHARACTERISTICS ACROSS COMPUTING TIERS.

Deployment Tier	Cloud LLMs (>175B)	Edge Servers (7B–70B)	On-Device SLMs (1B–3B)	Key Trade-offs
Model Parameters	175B+ (GPT-3/4 scale)	7B–70B (Llama3 scale)	1B–3B (Phi/TinyLlama)	Scale vs. deployability
Inference Latency	100ms–2s (network+compute)	50–200ms (local compute)	10–100ms (pure local)	Network dependency vs. speed
Power Consumption	100–1000W (data center)	20–100W (single GPU)	sub-10W (device)	Performance vs. efficiency
Memory Requirements	350GB–1TB+ (FP16 model+KV)	14GB–140GB (quantized)	<1MB–8GB (compressed)	Model capacity vs. constraints
Hardware Requirements	GPU clusters (H100/A100)	Mid-range GPU (A30/L4)	NPU/CPU/MCU	Capability vs. accessibility
Data Privacy	Network exposure	Local processing	Local processing	Capability vs. privacy
Connectivity Dependency	Always required	Intermittent	Optional	Robustness vs. capability
Deployment Complexity	Low (API-based)	Medium (local setup)	High (optimization)	Ease vs. customization

Note: Values represent typical implementation ranges. Actual performance varies by hardware configuration, optimization techniques, and workload characteristics.

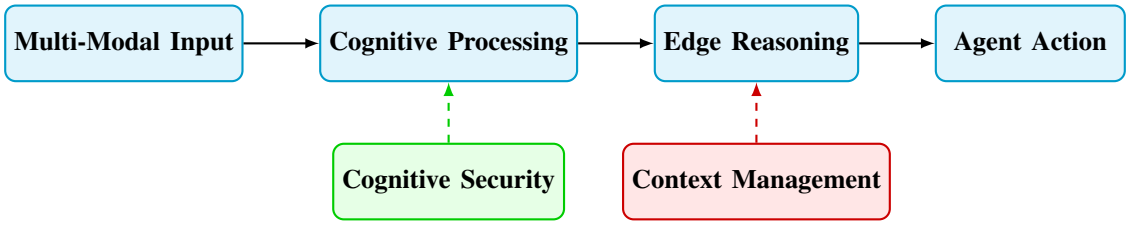
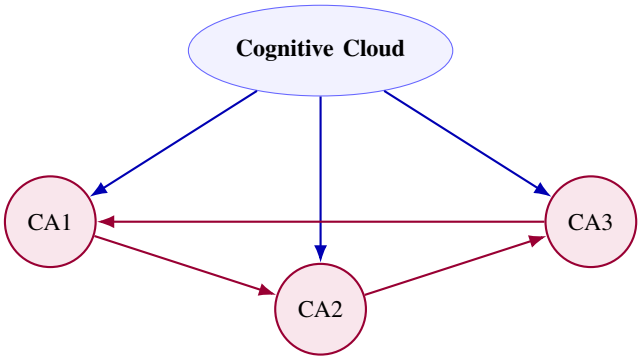


Fig. 4. Cognitive Edge Computing Pipeline: Multi-modal input processing through edge-based reasoning and autonomous agent action generation with cognitive security and context management.



Cognitive Agents (CA) collaborate via distributed reasoning and knowledge sharing.

Fig. 5. Distributed Cognitive Computing Architecture: Cognitive Agents collaborate through distributed reasoning, knowledge sharing, and cloud-assisted complex cognitive tasks.

response times [70]. Advanced quantization techniques (e.g., INT2/INT3) and sparse attention mechanisms enabled the execution of larger models (e.g., 20B+ parameters) on devices with just 8GB of memory [25]. Federated learning approaches [71] and collaborative inference frameworks like SLED [72] and DiSCo [73] enabled efficient, privacy-preserving personalization and computation offloading. The emergence of agent-native architectures marked the transition from static language models to truly autonomous edge agents capable of planning and tool use [74]. This progress was underpinned by dedicated AI accelerators achieving exceptional efficiency, enabling complex multimodal reasoning on edge devices [27].

B. Edge AI: Technical Definition and Constraints

Edge AI encompasses the deployment and execution of artificial intelligence algorithms directly on edge devices or edge computing infrastructure, positioned at the network periphery to minimize latency between data sources and processing units. This paradigm fundamentally shifts computational workloads from centralized cloud data centers to distributed edge nodes with significantly constrained resources.

Technical Characteristics and Quantitative Benefits (Indicative):

- **Ultra-Low Latency Processing:** Reported on-device inference can reach 1–50ms vs. 50–200ms with round-trip cloud latency for interactive tasks (modality and network dependent) [1], [2], [52].
- **Enhanced Privacy and Data Sovereignty:** Local processing retains sensitive data in situ aiding regulatory

alignment (GDPR/CCPA/HIPAA contexts) [1]; broad percentage reductions in attack surface are deployment specific so we avoid fixed universal values.
- **Bandwidth Optimization:** Feature / result transmission in lieu of raw streams can reduce upstream bandwidth by sizeable fractions (often tens of percent up to 90%) depending on compression and sampling strategies [2], [52].
- **Operational Resilience:** Local fallback mitigates intermittent connectivity; target availabilities of "three nines" are engineering goals rather than guaranteed universal outcomes.
- **Energy Efficiency:** Eliminating repeated radio transfers and exploiting low precision can yield multi-fold (often single- to low-double-digit) energy-per-inference improvements [55], [75]; higher outliers are workload- and hardware-specific.

These constraints necessitate the "optimization quad" (data, model, system, evaluation) approach to achieve viable edge AI deployment, transforming optimization from an enhancement strategy to a fundamental requirement for system feasibility.

C. Large Language Models: Architecture and Computational Requirements

Large Language Models (LLMs) represent a class of artificial neural networks, predominantly based on the Transformer architecture [9], trained on massive text corpora to achieve human-level language understanding and generation capabilities. These models have fundamentally transformed natural language processing through their emergent capabilities and scale-dependent performance characteristics.

Technical Architecture and Scale Characteristics:

- **Massive Parameter Scale:** Contemporary LLMs range from billions to hundreds of billions of parameters: GPT-3 (175B) [7], PaLM (540B) [10]; GPT-4 parameter count remains undisclosed [76]. Representative FP32 storage for disclosed scales spans hundreds of GB to multi-TB when including optimizer states.
- **Transformer-Based Architecture:** Multi-layer attention mechanisms with computational complexity $O(n^2d)$ for sequence length n and model dimension d, creating quadratic scaling challenges for long-context processing [9].
- **Generative Capabilities:** Auto-regressive text generation, reasoning, code synthesis, mathematical problem-solving, and multi-turn conversation with context windows ranging from 2K-1M+ tokens [44].
- **Emergent Abilities:** Scale-dependent capabilities including in-context learning, chain-of-thought reasoning, and few-shot task adaptation that emerge at specific parameter thresholds (typically >10B parameters) [8], [77].
- **Multimodal Extensions:** Integration with vision (CLIP, DALL-E), audio (Whisper), and other modalities, expanding input/output capabilities while increasing computational complexity [76]. Vision-language models face additional optimization challenges due to high-dimensional visual token sequences, requiring specialized approaches like FastViTHD encoders for efficient edge deployment [78].

Computational Resource Requirements: LLM deployment demands exceed typical edge device capabilities by multiple orders of magnitude [55], [75]:

- **Memory Requirements:** Large-scale models require memory proportional to parameter count (precision-dependent), with cloud-scale models demanding hundreds of GB to multi-TB storage capacity significantly exceeding typical edge device memory availability [7]
- **Inference Compute:** Real-time LLM inference requires computational throughput that substantially exceeds the processing capabilities available on standard edge hardware platforms [52]
- **Memory Bandwidth:** Efficient inference demands high-bandwidth memory access patterns that surpass the data transfer capabilities of resource-constrained edge devices [5]
- **Energy Consumption:** Cloud-scale inference power requirements significantly exceed the strict power budgets imposed by mobile and battery-powered edge deployment scenarios [2]

These computational requirements create the fundamental "deployment contradiction" that necessitates aggressive optimization strategies specifically designed for edge constraints [55], driving the development of Small Language Models (SLMs) and advanced compression techniques as essential pathways to edge viability [79], [80].

D. Cognitive Workload Characteristics

Cognitive workloads at the edge encompass a spectrum of tasks requiring advanced reasoning, multimodal processing, and autonomous decision-making under resource constraints. These workloads differ from traditional edge AI (e.g., image classification) by demanding:

- **Multi-step Reasoning:** Chain-of-thought processes for problem-solving, requiring sustained context across extended sequences that significantly exceed typical single-turn inference patterns [8].
- **Multimodal Integration:** Processing text, vision, and audio inputs simultaneously, as seen in models like CLIP, DALL-E, and Whisper [81]–[83].
- **Adaptive Planning:** Dynamic task decomposition and resource allocation based on environmental feedback [12].
- **Privacy-Sensitive Operations:** Local processing of personal data without cloud transmission, critical for healthcare and finance applications [1].

These characteristics necessitate edge-native architectures that preserve cognitive fidelity while operating within GB-scale memory and W-scale power envelopes.

E. AI Agents

AI Agents are autonomous software entities that perceive their environment, reason about complex scenarios, and execute goal-directed actions to accomplish specific objectives [12], [84]. In the context of cognitive edge computing, AI Agents

TABLE II
KEY CHARACTERISTICS AND FUNCTIONAL COMPONENTS OF AI AGENTS.

Feature/Component	Description	Purpose
Capabilities	Performs complex multi-step operations; learns and adapts; makes independent decisions; handles multimodal inputs	Enable sophisticated task execution
Interaction	Proactive and goal-oriented; communicates with other agents or humans when needed	Facilitate collaborative problem solving
Autonomy	Operates independently without constant human intervention; makes autonomous decisions	Reduce human workload
Reactivity	Perceives environmental changes and responds promptly	Maintain situational awareness
Proactiveness	Initiates actions and executes tasks to achieve objectives	Drive goal achievement
Social Skills	Communicates with other agents or humans	Enable collaboration
Reasoning and Planning	Analyzes data, identifies patterns, makes informed decisions; develops strategic plans	Support complex decision-making
Learning and Adaptation	Learns from experience, maintains context, adapts to new situations to improve performance	Enable continuous improvement
Functional Components	Data acquisition (sensors); processing and analysis (ML/AI); decision-making (algorithms/models); action execution	Core technical architecture
Types	Simple reflex; Model-based reflex; Goal-based; Utility-based; Learning agents	Categorize by capability level

represent the evolution from passive model inference to active, reasoning-capable systems that can adapt, plan, and collaborate within resource-constrained environments.

Modern AI Agents are characterized by four fundamental capabilities that distinguish them from traditional reactive systems [74], [85]: *autonomy* (independent operation without constant supervision), *reactivity* (responsive adaptation to environmental changes), *proactivity* (goal-oriented behavior initiation), and *social ability* (communication and collaboration with other agents or humans). These agents typically operate through a perception-reasoning-action cycle, incorporating data acquisition, intelligent processing, strategic decision-making, and action execution components [86].

The integration of LLMs as cognitive engines fundamentally transforms AI Agents from rule-based systems to sophisticated reasoning entities capable of natural language understanding, contextual planning, and adaptive behavior [13], [47]. This symbiotic relationship enables agents to transcend predetermined scripts and engage in complex, open-domain reasoning while maintaining edge deployment viability through recent advances in small language models optimized for agentic tasks [85]. Contemporary agent architectures span multiple complexity levels, from simple reflex agents to learning-enabled autonomous systems capable of multi-step planning and tool orchestration [14], [87].

Table II summarizes the key characteristics and architectural components that define modern AI agents in edge computing contexts, providing a structured framework for understanding their capabilities and deployment requirements.

F. Synergy: Edge AI with LLMs and Agents

The integration of Edge AI, LLMs, and AI Agents creates a powerful synergy for ubiquitous intelligence:

- **Real-time, Context-Aware Decision Making:** Edge deployment ensures that LLM-powered agents can react to local environmental changes instantaneously, critical for applications like autonomous vehicles [88] or industrial robots.
- **Enhanced Privacy and Security:** Processing LLM inference and agent logic on-device minimizes the transmission of sensitive user data to the cloud, adhering to privacy regulations and reducing attack surfaces [89].
- **Robust Autonomy:** Agents can maintain functionality even without continuous cloud connectivity, crucial for remote or intermittently connected environments [90].
- **Personalized and Adaptive Experiences:** LLM-powered agents can learn and adapt to individual user preferences or specific environmental conditions directly on the device, offering highly personalized services [91].
- **Distributed Intelligence:** Multi-agent systems at the edge can collaboratively solve complex problems by sharing local insights [92], reducing the burden on centralized cloud resources. Figure 5 illustrates the distributed cognitive computing architecture where cognitive agents collaborate through knowledge sharing and cloud-assisted complex tasks.

This synergy promises a future where intelligent systems are not only pervasive but also highly responsive, secure, and adaptable to dynamic real-world scenarios.

III. CHALLENGES IN DEPLOYING EDGE LLMS AND AI AGENTS

Building on the foundational understanding of cognitive edge computing components, this section examines the deployment challenges that arise from the fundamental mismatch between the resource requirements of advanced AI systems and the constraints of edge environments. The integration of LLMs and AI Agents with edge computing creates unprecedented challenges that fundamentally exceed traditional Edge AI

constraints by 2-3 orders of magnitude, requiring revolutionary optimization approaches.

A. Fundamental Limitations of Cloud-Centric AI Deployment

Before examining technical optimization challenges, we identify three critical limitations of cloud-dependent AI deployment that drive the imperative for edge-native solutions:

- **Network Dependency and Connectivity Constraints:** Cloud-based AI services suffer from fundamental connectivity dependencies that render them unusable in disconnected environments. Round-trip latencies of 50-500ms to cloud endpoints [52] are inadequate for real-time applications requiring <10ms response times [5]. Network outages, poor connectivity in rural areas, underground facilities, aircraft, and maritime environments completely eliminate AI capability access. Edge scenarios with privacy-sensitive, latency-critical, or bandwidth-constrained requirements (medical devices, autonomous vehicles, industrial control systems) cannot tolerate cloud dependencies [4], [5].
- **Privacy and Data Sovereignty Concerns:** Cloud processing necessitates uploading sensitive user data (conversations, documents, biometric information, location data) to remote servers, creating privacy vulnerabilities and regulatory compliance challenges. GDPR, HIPAA, and other data protection regulations restrict cross-border data transfer, limiting cloud AI deployment in regulated industries [5]. Corporate and government environments require data to remain within controlled boundaries, prohibiting cloud-based AI processing for classified or proprietary information [6].
- **Limited Personalization and Context Adaptation:** Cloud models serve global user populations, constraining deep personalization to individual user patterns, linguistic preferences, domain-specific knowledge, and contextual behaviors. Continuous adaptation based on user interactions requires persistent model fine-tuning, which is computationally prohibitive and privacy-compromising when performed in cloud environments. Local context (device usage patterns, environmental sensors, personal preferences) cannot be effectively integrated into cloud-based decision making without extensive data transmission [4], [93].

These limitations provide strong motivation for edge-native AI deployment, despite the significant technical challenges outlined in subsequent sections.

B. Quantified Resource Constraint Analysis

- **Computational Power Mismatch:** Typical interactive LLM inference may demand on the order of tens to several hundred GFLOPS per token depending on architecture and sequence length, whereas many edge devices expose only low single-digit to tens of GFLOPS sustained [7], [55]. Additional agent perception and planning stages further compound this gap [86]. Recent approaches like APEX demonstrate how hybrid CPU-GPU execution and optimized scheduling can improve throughput by 11-96% on constrained hardware through better resource utilization [94].
- **Memory Footprint Crisis:** Large-scale LLMs create severe memory constraints for edge deployment. GPT-3, with 175 billion parameters at 16-bit precision, requires 350GB of storage [7], while even modest 10B parameter models demand up to 20GB of main memory (DRAM) with INT8 quantization [95]. In contrast, edge devices offer limited memory: high-end smartphones provide 6-12GB DRAM (e.g., iPhone 15 with 6GB), while commodity devices operate with even tighter constraints [95]. This creates a deployment gap where quantized models still exceed edge memory by 2-10×. AI Agents compound this by requiring additional memory for environmental states, historical context, planning graphs, and multimodal data buffers [12]. Emerging solutions include SLED for model sharing across devices [72], adaptive quantization techniques like QPART [96], and collaborative frameworks such as EdgeShard for distributed model partitioning [22].
- **Energy Consumption Bottleneck:** LLM inference often exceeds the power budgets of mobile and edge devices, necessitating sophisticated optimization strategies that combine precision reduction, sparsity, and intelligent scheduling [50]. The energy disparity across computing tiers is substantial, as illustrated in Figure 6, with cloud data centers consuming MW-scale power for facility-scale operations, while edge servers operate at tens of watts and mobile devices consume sub-10W [57], [59], [97]. This multi-order-of-magnitude gap highlights the critical need for energy-efficient optimization techniques. Agentic processes, incorporating sensorimotor loops and continuous planning, introduce additional energy demands that must be carefully managed. Research in federated multi-agent reinforcement learning (Fed-MARL) demonstrates promising approaches for energy-aware resource management in 6G edge networks, optimizing latency, energy efficiency, and reliability under stringent constraints [98].
- **Memory Bandwidth Limitations:** High-throughput LLM inference can saturate memory bandwidth, creating significant bottlenecks in auto-regressive generation where weights and activations are repeatedly accessed [50]. The bandwidth disparity is substantial: data center GPUs like NVIDIA A100 offer 1555 GB/s bandwidth, while mobile SoCs typically provide 50-60 GB/s (e.g., Snapdragon 8 Gen 2) [50]. This order-of-magnitude difference results in multi-fold slowdowns during token generation on edge devices. Frameworks like APEX address these limitations through parallel CPU-GPU execution and optimized attention computation offloading, improving throughput by 84-96% on constrained GPUs [94]. Similarly, EdgeShard's collaborative approach partitions models across devices, effectively distributing memory bandwidth requirements and reducing latency by up to 50% [22].
- **Communication Infrastructure Constraints:** Edge networks typically provide significantly lower bandwidth compared to data center interconnects, creating substantial

[5] https://aws.amazon.com/what-is/rtt-in-networking/

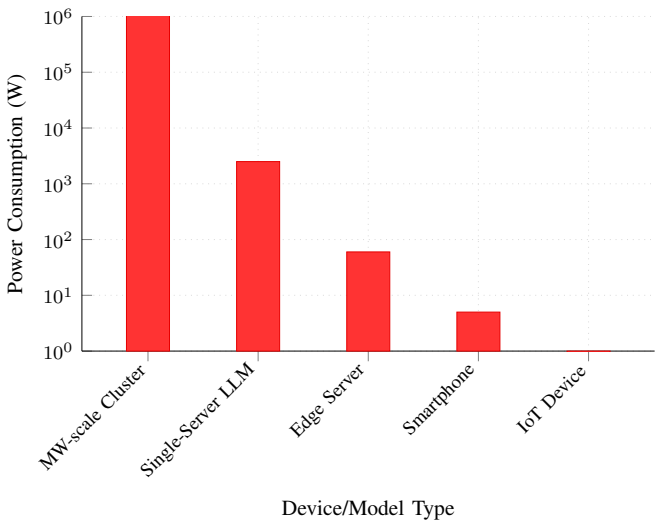


Fig. 6. Power consumption disparity across computing tiers (log scale). Representative devices: MW-scale Cluster: Huawei Atlas 950 SuperCluster (10MW facility with cooling) ; Single-Server LLM: High-end server with 2×A100 80G GPUs ; Edge Server: NVIDIA Jetson AGX Orin (30-60W) ; Smartphone: Modern flagship smartphone (e.g., iPhone 17 or Samsung Galaxy S25 Ultra) ; IoT Device: ESP32-based microcontroller [57], [59], [97]. Facility-scale includes cooling and infrastructure overhead. Per-query energy (Wh/query) varies significantly with latency, sequence length, and optimization techniques as discussed in Section III-E.

bottlenecks for federated learning and collaborative inference scenarios [50]. This constraint profoundly impacts scenarios where transmitting parameter updates can cause communication delays ranging from minutes to hours over constrained edge networks. Innovative approaches like pFL-SBPM demonstrate communication-efficient personalized federated learning, reducing uplink communication costs by 96.875% through random binary probability masks instead of transmitting full precision weights or gradients [99]. For latency-sensitive applications, EdgeShard employs collaborative edge computing to minimize data transmission, supporting full-precision model inference without accuracy loss [22].

C. Technical Deployment and Management Complexities

- **Hardware Heterogeneity Challenge:** Edge ecosystems encompass numerous distinct hardware architectures with varying instruction sets (ARM, x86, RISC-V), accelerators (CPU, GPU, NPU, TPU), and memory hierarchies [5]. Optimizing LLMs for this heterogeneity requires: 1) Platform-specific compilation using dozens of compiler toolchains; 2) Architecture-aware model partitioning strategies; 3) Dynamic resource allocation algorithms; 4) Cross-platform performance optimization, approaching platform-optimal performance.
- **Dynamic Adaptation and Learning Overhead:** Full on-device fine-tuning can often require an order of magnitude or more compute relative to inference; parameter-efficient approaches (e.g., LoRA [100], adapters [101]) substantially lower but do not eliminate the gap. Continuous adaptation cycles must therefore be scheduled opportunistically to balance learning benefits with resource constraints [102].
- **Multi-Model Orchestration Complexity:** AI Agents often require coordination of several to dozens of specialized models (vision, language, planning, control), each with different resource requirements and inference patterns [103]. Cold start latencies can range from hundreds of milliseconds to several seconds, while task switching may introduce delays from tens to hundreds of milliseconds, critically impacting real-time performance requirements in latency-sensitive applications [104].

Edge scenarios with privacy-sensitive, latency-critical cognition—such as vehicular navigation requiring <10ms response times for obstacle avoidance, clinical triage in remote areas without network connectivity, industrial diagnostics for predictive maintenance, biomarker-driven screening in point-of-care devices, and civic infrastructure monitoring for anomaly detection—drive the need for local cognitive processing to ensure personalization, resilience, and compliance [5], [6], [60], [61].

D. Security, Privacy, and Trustworthiness Challenges

- **Privacy-Preserving Computation Overhead:** Differential privacy, secure aggregation, and homomorphic encryption introduce non-trivial (sometimes prohibitive) latency and communication overhead; practical deployments typically balance privacy budgets against real-time constraints [105].
- **Attack Surface Expansion:** Distributed edge AI broadens potential attack vectors compared to centralized deployments. Representative vectors include model extraction attacks requiring thousands to millions of queries, adversarial input crafting that often achieves high success rates on undefended models, Byzantine attacks in federated learning that can potentially affect a significant portion of participating nodes, and physical tampering of edge devices that may comprise a notable portion of deployed systems [106].
- **Autonomous Agent Safety Risks:** AI Agents operating in physical environments pose quantifiable safety risks. These include decision latency failures where delays of several milliseconds can cause safety violations, varying hallucination rates for complex reasoning tasks, varying out-of-distribution detection accuracy for novel scenarios, and varying multi-agent coordination failure rates in distributed systems [98].
- **Explainability and Verification Challenges:** LLM decision paths involve tens of thousands to millions of computational steps, making complete verification computationally intractable. Current explanation methods achieve typically are insufficient for safety-critical applications requiring extremely high reliability [107].

E. Energy Metrics and Reporting

For cross-tier comparisons we distinguish between: (i) instantaneous device power (W), (ii) facility power including cooling/overhead (kW–MW), and (iii) energy per request

measured in Wh/query. The latter depends on end-to-end latency (prompt length, generation length), precision (e.g., INT4–INT8), and utilization. We recommend reporting both steady-state TDP and measured Wh/query with workload description and hardware configuration, following energy-aware evaluation practices in edge NLP and mobile AI challenges [108], [109].

Lifecycle and Sustainability Considerations. Beyond runtime energy, lifecycle analysis (LCA) should consider hardware manufacturing/refresh cycles, thermal aging, and software stack updates. We recommend (i) reporting the functional unit (tokens generated per Joule over device lifetime), (ii) separating embodied vs operational energy, and (iii) documenting e-waste mitigation (e.g., model compression extending device lifetime). Where possible, align with Green AI reporting practices and include Scope 2/3 boundary notes.

F. Detailed Technical Constraint Analysis for Edge-Side Large Models

The deployment of edge-side large models faces several critical technical constraints that fundamentally limit their practical implementation [6], [110]–[112]:

1. Computational Performance Limitations: Edge devices provide limited computing capacity and memory bandwidth compared to cloud infrastructure, creating significant performance bottlenecks for LLM inference due to the memory-bound nature of transformer architectures.

2. Power Consumption and Thermal Management: Running LLMs on edge devices consumes several watts, often exceeding typical power budgets and causing thermal throttling that significantly reduces performance and creates unpredictable latency.

3. Model Quantization Trade-offs: Quantization from higher precision to INT8/INT4 or lower bit widths introduces accuracy degradation, particularly on reasoning tasks, requiring careful calibration and mixed-precision strategies to balance performance and quality.

4. Immature Development Ecosystem: Converting models to edge-optimized formats requires extensive manual adjustments due to immature toolchains and limited profiling capabilities, hindering systematic optimization efforts.

5. Platform Fragmentation: Hardware and software heterogeneity across vendors and platforms necessitates vendor-specific optimizations, limiting model portability and increasing development complexity.

6. Context Length Limitations: Extended context windows severely impact performance due to quadratic memory scaling in attention mechanisms, requiring sophisticated cache management for multi-turn conversations.

With a clear understanding of the challenges in deploying LLMs and AI agents at the edge, the next section explores the optimization strategies that can address these issues through coordinated approaches across data, model, and system levels.

IV. Optimization Strategies for Edge LLMs and AI Agents

Figure 7 presents a comprehensive Edge AI optimization ecosystem, illustrating how cloud-based LLMs are progressively optimized through multi-layer techniques—including data, model, and system-level strategies—before deployment to diverse edge devices. The diagram highlights the flow and interaction between optimization stages, demonstrating how coordinated improvements across the stack enable cognitive computing at the edge and deliver quantified performance gains, providing a global perspective for understanding the subsequent detailed optimization methods. As discussed in the previous section, deploying LLMs and AI Agents on edge devices faces significant challenges related to computational resources, energy consumption, and system complexity. To mitigate these challenges, a holistic approach leveraging optimization strategies across data, model, and system levels is crucial, encompassing techniques such as quantization (INT8/INT4 (8-bit/4-bit integer) precision reduction achieving 4-8$\times$ compression), pruning (structured and unstructured approaches for model size reduction), knowledge distillation (KD) (teacher-student compression), architecture design innovations (purpose-built Small Language Models and efficient attention mechanisms), and system-level optimizations (model partitioning, hardware acceleration, and dynamic scheduling). The effectiveness of these techniques varies based on model architecture, hardware platform, and application requirements, with implementation complexity ranging from straightforward quantization to sophisticated multi-teacher distillation frameworks [5], [6].

A. Data Optimization

Data optimization techniques focus on preparing and enhancing datasets to improve the efficiency and effectiveness of edge-deployed LLMs and AI agents. These techniques address the unique challenges of edge environments, including limited storage capacity, privacy constraints, and the need for high-quality training data that can be processed with minimal computational resources.

Figure 8 illustrates the comprehensive framework for processing both edge data and cloud training data through multiple optimization strategies.

- **Data Cleaning and Preprocessing:** High-quality local datasets mitigate noise and hallucination risks [113]. Active label or federated cleaning strategies reduce redundant transmission while preserving privacy [6].
- **Feature Compression:** Dimensionality reduction and embedding compression target context transfer bottlenecks between edge and cloud [6], [63]. Techniques such as Principal Component Analysis (PCA), autoencoders, and feature hashing are employed to extract salient features, thereby compacting data while preserving essential information relevant for edge models.
- **Data Augmentation:** Synthetic and transformation-based augmentation enlarges scarce edge task corpora; combined with KD it enhances student specialization [6], [114].
- **Mitigating Bias from Synthetic Data:** Synthetic augmentation may import distributional artifacts; auditing and mixed real–synthetic sampling reduce bias risks [115].
  - **Differentially Private Synthetic Data:** This approach mimics the statistical patterns of real data without containing personally identifiable information

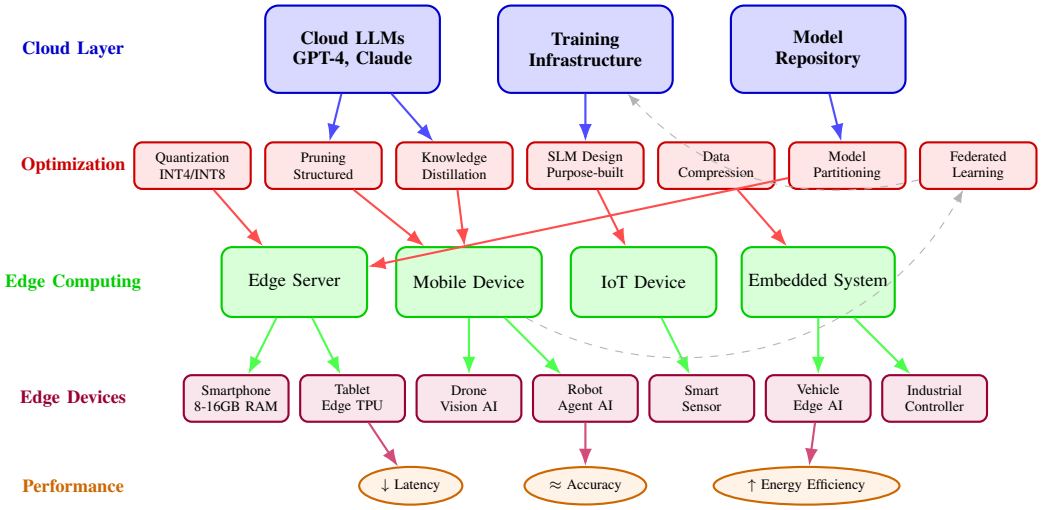


Fig. 7. Comprehensive Edge AI Optimization Ecosystem: End-to-end system architecture showing the flow from cloud LLMs through optimization techniques to edge deployment, demonstrating how multi-layer optimization enables cognitive computing across diverse edge devices while achieving quantified performance improvements.

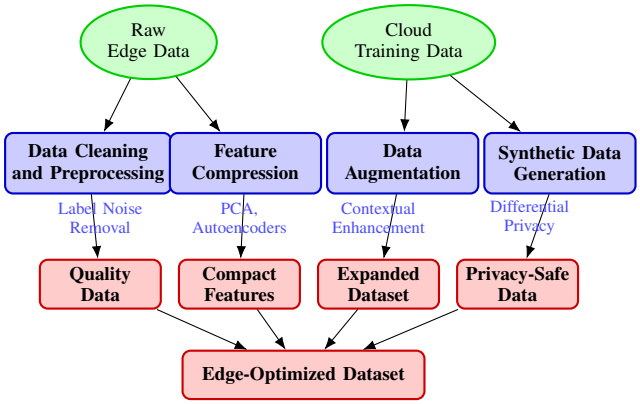


Fig. 8. Data Optimization Techniques for Edge AI Deployment. The framework processes raw edge data and cloud training data through multiple optimization strategies to create edge-suitable datasets.

(PII), enabling data expansion, sharing, and reuse while minimizing privacy leakage risks [115].

– **Feedback-Driven Augmentation:** This method iteratively improves synthetic data generation by incorporating user interactions, domain expert input, and system performance metrics, ensuring that synthetic data remains relevant, representative, and aligned with system requirements [116]. This approach helps generate more balanced datasets, reducing public data bias and under-representation issues, particularly in data-limited domains such as fraud detection, medical research, or recruitment [115].

B. Model Optimization

Model optimization focuses on adapting the AI model itself to be more resource-efficient without significant loss in performance. This is particularly critical for LLMs, given their inherent size. We summarize typical trade-offs between compression techniques (quantization, pruning, KD, low-rank, sharing), noting accuracy–efficiency frontiers vary by task/hardware [55], [75], [125].

1) Practical density heuristics for edge-side model design: Deploying large language models at the edge is ultimately constrained by compute, memory bandwidth, latency, and power budgets. Rather than proposing a new "law," we adopt a pragmatic view: density describes the amount of useful work delivered per unit of constrained resource (Joule, byte/s, or ms). Empirical studies consistently show that transformer inference at the edge is frequently memory-bandwidth-bound and thermally limited on mobile SoCs, with end-to-end performance sensitive to precision, caching, and scheduling choices [2], [43], [52], [55].

Observed patterns and actionable heuristics:

- **Memory/bandwidth awareness:** Sustained throughput depends more on data movement than on peak TOPS. Favor precision schemes and layouts that minimize bandwidth pressure (e.g., weight-only or mixed-precision quantization, KV-cache locality, prefetch-friendly packing) and align with the target memory hierarchy [43], [55].
- **Power/thermal budget as a first-class constraint:** On battery-powered devices, thermal throttling can dominate steady-state throughput. Runtime policies (duty-cycling, burst scheduling), mixed precision, and operator fusion help preserve efficiency under thermal limits [2], [126], [127].
- **Scale within hardware envelopes:** Small-to-mid scale models (from sub-1B up to the low tens of billions of parameters on edge servers) are commonly reported for on-device and near-edge use; the practical breakpoint depends on task, latency, and memory budgets [43]. Architectural tailoring (efficient attention, compact FFNs) generally yields better density than naïve downscaling [55].
- **Collaborative execution matters:** In edge–cloud or multi-edge settings, overall latency and throughput are bounded by the slower of compute and the communication fabric. Partitioning and offloading must account for link variability; hiding communication behind compute and compressing activations/KV state are often decisive [27], [43].

TABLE III
COMPARISON OF SMALL LANGUAGE MODEL (SLM) ARCHITECTURES FOR EDGE AI.

SLM Name	Architecture Type	Key Features and Innovations	Parameters and Size Reduction	Reported Performance	Edge Suitability and Use Case
MobileBERT [117]	Encoder-only	Inverted bottleneck structure, balances attention layers	4.3× smaller, 5.5× faster	Near-BERT performance	Mobile devices
DistilBERT [118]	Encoder-only	Knowledge distillation	>60% smaller, 96% BERT	Highly efficient	Resource-limited
TinyBERT [119]	Encoder-only	Distillation + augmentation	Compact, 96% BERT	Production-ready	Constrained envs
BabyLLaMA [120]	Decoder-only	Multi-teacher distillation	58M params	Low-data performance	Low-data devices
TinyLLaMA [121]	Decoder-only	FlashAttention	1.1B params	Memory efficient	Memory-limited
MobileLLM [68]	Decoder-only	Weight sharing, GQA	Low latency	Practical deployment	Mobile systems
LLaMA 3.1 8B [122]	Decoder-only	Compact LLM	8B params	Fine-tunable	LLM-capable edge
Pythia [123]	Decoder-only	Interpretability	160M-2.8B	Benchmarking	Research use
SmolLM2-1.7B [124]	Decoder-only	Curated datasets	1.7B params	Task-efficient	Specific NLP tasks

Design implications: These observations translate into practical guidance rather than strict formulas:

- **Architecture selection:** Prefer sparse/efficient attention and lean feed-forward blocks that maximize useful computation per byte moved [55].
- **Quantization strategy:** Use aggressive but validated quantization where supported by the toolchain, while maintaining precision where sensitivity is high (e.g., attention, logits) [43], [55], [128], [129].
- **Memory hierarchy optimization:** Co-design kernels and layouts to reduce DRAM traffic and exploit caches/KV locality; batch/window sizing should follow bandwidth, not just FLOPs [130].
- **Adaptive scheduling:** Adjust concurrency, precision, and offload boundaries in response to runtime resource and thermal telemetry [2], [27].

We use the term "density" heuristically in this survey. The above patterns synthesize reported behavior across devices and workloads; concrete thresholds (e.g., tokens/s or power draw) are deployment- and hardware-specific and should be established via measurement on the target platform [52].

- **Compact Architecture Design (Small Language Models - SLMs):** Table III compares representative small language model architectures and their edge suitability. SLMs are specifically designed as lightweight architectures with inherently lower computational and memory requirements compared to large models [113], [131], [132]. Their parameters typically range from millions to billions, representing a significant reduction compared to LLMs with hundreds of billions of parameters [113], [131].
Advantages: SLMs offer numerous advantages for edge deployment, including faster deployment cycles, easier fine-tuning on proprietary data, lower energy consumption, higher sustainability, and natural suitability for resource-constrained environments such as mobile devices, embedded systems, and edge hardware [85], [113], [131], [132]. Recent SLMs demonstrate lower computational requirements and faster inference compared to larger models, with specialization for domains like healthcare, legal, and supply chain applications [45], [66], [133]–[135]. From an enterprise and agentic AI operations perspective, SLMs can deliver lower latency, cost/energy, and stronger privacy/control when embedded at the edge, enabling safe iteration and workflow integration at scale.
Modern multilingual encoders (for retrieval/embedding at the edge): While decoder-only LLMs dominate generation, encoder-only models remain critical for multilingual retrieval, classification, and embedding services that front many edge pipelines. The recent mmBERT revisits multilingual encoders with a modern recipe, reporting 2–4× inference speedups over XLMR with state-of-the-art performance on XTREME and MTEB / CoIR [136]. Such encoders can serve as fast, memory efficient front ends for on device search, re-ranking, and agent tools in multilingual settings.
- **Advanced Architecture Innovation for Edge Deployment:**
MobileLLM Deep-Narrow Architecture Design: Mo-

bileLLM introduces a revolutionary approach to sub-billion parameter language models through deep-narrow architectural optimization [68]. The design principle prioritizes depth over width, achieving superior parameter efficiency by employing deeper transformer layers with reduced hidden dimensions. This approach demonstrates that deep-narrow architectures can achieve performance equivalent to wider models while requiring significantly fewer computational resources during inference. The MobileLLM framework achieves notable improvements in edge deployment scenarios through optimized depth-width trade-offs, immediate block-wise computation, and efficient attention pattern utilization.

MobileLLM-R1 Efficient Inference Series: Building on the original MobileLLM architecture, Meta AI's MobileLLM-R1 series advances efficient inference for small language models [137]. The series includes base models (MobileLLM-R1-140M/360M/950M) and SFT variants specialized for mathematics, programming, and scientific reasoning. Reported evaluations indicate the 950M model (trained on 2T high-quality tokens; total <5T) attains higher scores than Qwen3 0.6B on MATH, GSM8K, MMLU, and LiveCodeBench under the authors' specified setups, and competitive coding performance among open models. Meta provides training recipes and data sources to support reproducibility.

EdgeShard Collaborative Edge Computing: EdgeShard represents a breakthrough in distributed LLM inference through collaborative edge computing architectures [22]. The system partitions large language models across multiple edge devices, enabling coordinated inference that achieves reduced latency and improved throughput compared to single-device deployment. EdgeShard's innovations include intelligent model partitioning algorithms, communication-efficient synchronization protocols, and adaptive load balancing strategies that optimize resource utilization across heterogeneous edge infrastructure. The framework demonstrates significant scalability improvements, allowing deployment of larger models through resource aggregation while maintaining edge computing advantages.

Mixture-of-Experts (MoE) Edge Optimization: Edge-optimized MoE architectures, including EdgeMoE, LocMoE, and JetMoE variants, adapt sparse expert routing for resource-constrained environments [23], [43], [138]–[141]. These approaches reduce computational overhead through selective expert activation while maintaining model capacity, enabling deployment of sophisticated reasoning capabilities within edge device constraints. Key innovations include dynamic expert pruning based on device capabilities, hierarchical expert organization for memory efficiency, and context-aware routing that optimizes expert selection for specific deployment scenarios. The sparse activation patterns typical in MoE architectures align well with edge computing requirements, providing computational efficiency gains of 2-5$\times$ compared to dense architectures while preserving model quality.

Collaborative Multi-Device Inference Patterns: Advanced deployment strategies leverage multiple edge devices for coordinated inference, distributing computational load across smartphone clusters, IoT device networks, and edge server infrastructures. These patterns include pipeline parallelism for sequential transformer layers, tensor parallelism for attention computation distribution, and hybrid approaches that adapt to dynamic resource availability and network conditions.

- **Parameter-Efficient Fine-Tuning (PEFT) for Edge Deployment:**
  Low-Rank Adaptation (LoRA)-based Parameter-Efficient Fine-Tuning represents a breakthrough approach enabling on-device model adaptation with minimal computational overhead [142]. This methodology addresses the critical challenge of personalizing large language models on resource-constrained edge devices without requiring full parameter updates.

  Theoretical Foundation and Optimization Efficiency: Recent theoretical analysis demonstrates that fine-tuning attention mechanisms through selective parameter adaptation achieves superior generalization bounds while maintaining memory efficiency [142]. The approach leverages information-theoretic generalization bounds, proving that fine-tuning only query, key, and value matrices with identical rank constraints can achieve performance equivalent to or better than full parameter fine-tuning while reducing parameter count and improving generalization limits.

  On-Device Memory and Time Optimization: Advanced PEFT implementations achieve 20-40% memory reduction during edge training while maintaining or improving learning effectiveness [143], [144]. This efficiency enables practical on-device personalization scenarios including custom assistant training, photography algorithm optimization, and user-specific preference adaptation directly on mobile devices without cloud dependency.

  Learning Dynamics and Convergence Analysis: Theoretical insights reveal that asymmetric learning rates in attention mechanism fine-tuning, where query matrix learning rates significantly exceed key-value matrix rates, enable more efficient feature learning [142]. This principle guides practical implementations across full fine-tuning, LoRA, and DoRA methodologies, demonstrating orthogonal compatibility with different fine-tuning approaches.

  Edge Applications and Deployment Scenarios: PEFT enables diverse edge deployment scenarios including: (1) Vertical domain knowledge enhancement for medical, legal, and financial applications requiring specialized accuracy; (2) Task-specific optimization for customer service, summarization, writing assistance, and sentiment analysis; (3) User preference adaptation based on historical behavior patterns and personalized service requirements. The reduced resource requirements make sophisticated AI personalization accessible on consumer mobile devices.

  Examples: Architectures are broadly categorized into encoder-only architectures (e.g., MobileBERT, DistilBERT, TinyBERT, which achieve significant size reduction while maintaining performance [145]) and decoder-only archi-

tectures (e.g., BabyLLaMA, TinyLLaMA, MobileLLM [145]). Recent multimodal SLMs include Apple's MobileCLIP2, which achieves competitive performance with half the parameters of comparable models through multi-modal reinforced training . Meta's LLaMA 3.1 8B is highlighted as a compact model that retains significant LLM-level capabilities [113], [131].

Instead of deploying colossal LLMs, a growing trend is to design intrinsically smaller models (SLMs) that are optimized for edge inference. These models typically have millions or billions of parameters, significantly less than their cloud-based counterparts, while retaining strong performance on specific tasks. Examples include MobileBERT [117], DistilBERT [118], TinyBERT [119], BabyLLaMA, TinyLLaMA, Microsoft's Phi [45], Google's Gemma [133], Apple's OpenELM [66], and IBM's Granite [134]. Recent developments also include smaller versions of powerful LLMs like LLaMA 3.1 8B [146] specifically designed for on-device inference. For Transformer architectures, this involves designing lightweight attention mechanisms (e.g., sparse attention, linear attention) and reducing the number of layers or hidden dimensions while maintaining critical functionality [131].

- **Neural Architecture Search (NAS):** NAS automates neural architecture exploration under multi-objective constraints (accuracy, latency, energy) [6], [147]–[149]. Figure 9 illustrates the comprehensive NAS framework for discovering optimal architectures that balance performance with resource constraints through reinforcement learning and evolutionary algorithms.

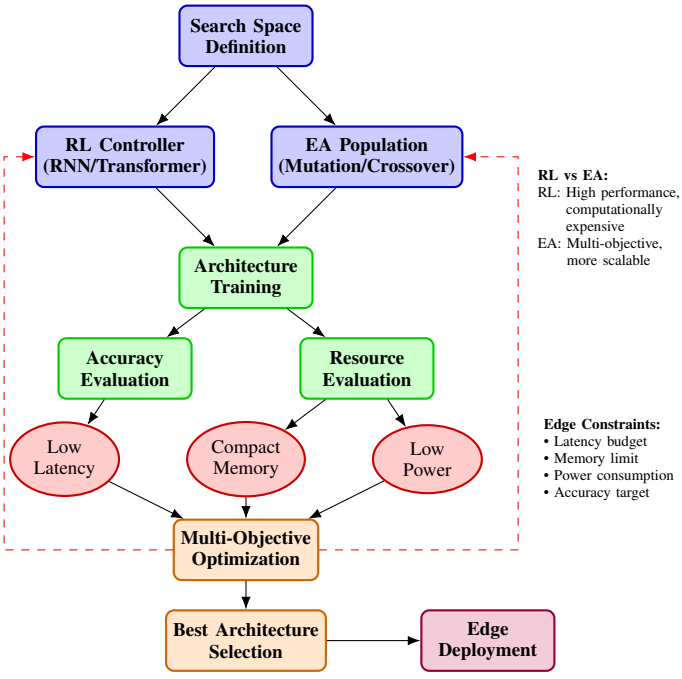


Fig. 9. Neural Architecture Search (NAS) Framework for Edge AI. The system automatically discovers optimal architectures balancing accuracy and resource constraints through reinforcement learning or evolutionary algorithms.

Advanced NAS variants incorporate runtime intermittency, reliability, and federated non-IID constraints (e.g., resource-aware, fault-tolerant search) [6].

Trade-offs between Reinforcement Learning and Evolutionary Algorithms: NAS typically employs reinforcement learning (RL) or evolutionary algorithms (EA) to explore the vast architecture space [147], [150]–[152].

- **Reinforcement Learning (RL) Methods:** These methods typically use controllers (such as RNNs or Transformers) to generate architectures and receive reward signals based on performance metrics (such as accuracy, latency) [150], [151]. RL excels at finding high-performance architectures, especially in large search spaces [150]. However, RL methods can be computationally expensive and inefficient in resource-constrained environments [150].
- **Evolutionary Algorithm (EA) Methods:** EAs simulate natural selection processes, evolving candidate architectures through mutation, crossover, and selection [147], [150], [152], [153]. Multi-task and high-dimensional search efficiency advances further reduce exploration cost on resource-constrained platforms [148]. EAs have advantages in multi-objective optimization, simultaneously balancing accuracy, resource and power consumption [147], [148], [150], and remain scalable relative to RL in constrained settings.
- **Trade-offs:** In resource-constrained edge environments, the choice between RL and EA depends on specific requirements. RL may be more effective in finding optimal performance but is computationally more expensive; while EA may perform better in efficiency and scalability, especially when balancing multiple constraints [150]. For example, resource-efficient methods like Random Search with weight-sharing can even outperform complex NAS algorithms on certain benchmarks, indicating that simple, random methods may also be competitive in resource-constrained edge AI [147].

- **Model Compression Techniques:** These techniques reduce the size and computational complexity of pre-trained models, often referred to as the "three fundamental approaches" for model compression. For LLMs, the following are especially relevant:
  - **Network Pruning (Trimming the Garden):** Eliminates redundant weights, connections, or even entire neurons/layers from a trained network, analogous to pruning a garden by removing unnecessary branches while preserving the essential structure. *Song Han et al.'s work on Deep Compression* [154] demonstrated significant size reductions (35-49x) for CNNs by combining pruning, quantization, and Huffman coding. For LLMs, pruning can target specific components like redundant attention heads, entire Transformer layers (depth-pruning), or individual neurons/connections within feed-forward networks (width-pruning). Recent works such as LLM-Pruner [155] and SparseGPT [156] have enabled structured and unstructured pruning for billion-parameter LLMs with minimal accuracy loss [157]. Pruning can be structured (removing

entire blocks, easier for hardware acceleration) or unstructured (removing individual weights, higher compression but harder to accelerate). Post-pruning fine-tuning is often necessary to recover accuracy.

– **Parameter Sharing:** Forces different parts of the model to share the same weights. This can be applied within or across Transformer layers in LLMs, significantly reducing the total number of unique parameters. Representative works include ALBERT [158] and MobileLLM [68].
– **Speculative Sampling for Accelerated Inference:** Advanced techniques that accelerate auto-regressive generation through draft-then-verify mechanisms, enabling multiple tokens per forward pass while maintaining output quality equivalent to standard decoding:

  * **FR-Spec: Frequency-Ranked Speculative Sampling:** A revolutionary approach addressing efficiency challenges in large-vocabulary language models through vocabulary space compression [159]:
    Core Innovation: FR-Spec constrains draft model search to frequency-prioritized token subsets, reducing LM Head computation overhead by 75% while ensuring equivalence of final output distribution. This addresses the critical bottleneck where large vocabularies (e.g., Llama-3-8B with 128k tokens) significantly impact speculative sampling efficiency.
    Vocabulary Optimization Strategy: Analysis reveals that 75% of tokens in large model vocabularies contribute less than 5% of total occurrence frequency. FR-Spec exploits this distribution by dynamically constraining draft candidate selection to high-frequency token subsets, achieving average 1.12× speedup over state-of-the-art EAGLE-2 methods.
    Lossless Acceleration Guarantee: Unlike pruning-based approaches, FR-Spec maintains complete output distribution equivalence by constraining only draft generation while preserving full vocabulary during verification phases. This ensures quality preservation while achieving substantial computational savings.
  * **EAGLE-2: Enhanced Tree-Attention Architecture:** Advanced tree-based speculative decoding with sophisticated attention mask management for complex draft tree structures [160]:
    Tree-Based Draft Generation: EAGLE-2 constructs speculative trees from given prefixes, generating multiple draft paths and taking path unions to form comprehensive draft trees. This approach enables parallel processing of multiple generation hypotheses within single forward passes.
    Optimized Attention Mask Implementation: CPM.cu's implementation addresses FlashAttention limitations for speculative sampling through compressed attention masks. Traditional int32 masks are compressed to uint64 representations for tree sizes up to 64 tokens, maintaining performance parity while supporting complex attention patterns.
    Shared Memory Management: Advanced preloading strategies transfer compressed attention masks to GPU shared memory, eliminating global memory access bottlenecks that traditionally degraded FlashAttention performance during speculative decoding operations.
  * **MTP: Multi-Token Prediction Architecture:** Lightweight draft models optimized for speculative sampling with minimal computational overhead [161]:
    Single-Layer Design Philosophy: MTP utilizes minimal architecture comprising single transformer layer plus language modeling head, achieving extreme efficiency while maintaining effective draft generation capabilities for edge deployment scenarios.
    LM Head Optimization Challenge: Implementation reveals that vocabulary size significantly impacts small model efficiency, with LM Head computation becoming the primary bottleneck. For models with vocabulary sizes orders of magnitude larger than hidden dimensions, LM Head operations dominate total computation time.
    Edge Device Performance Characteristics: MTP's minimal architecture proves particularly suitable for edge deployment where memory bandwidth and computational resources are severely constrained, enabling efficient speculative sampling on mobile and embedded platforms.
  * **SpecMQuant: Speculative Sampling with Quantization Integration:** Comprehensive framework combining speculative decoding with quantization strategies for maximum edge efficiency [162]:
    Hierarchical Framework Design: SpecMQuant establishes compatibility evaluation protocols between speculative sampling and quantization techniques, addressing the challenge of combining multiple acceleration strategies without performance degradation.
    Quantization-Aware Speculative Training: Advanced training methodologies that simultaneously optimize models for both quantization robustness and speculative sampling effectiveness, achieving compound acceleration benefits while preserving output quality.
    Hardware Co-Design Integration: Framework design considerations for hardware-specific quantization formats (int4, int8) combined with speculative sampling patterns, optimizing both memory access patterns and computational efficiency for edge NPU architectures.
  * **EdgeShard Collaborative Inference:** Distributed LLM inference through collaborative edge computing architectures that partition large models

across multiple edge devices, achieving reduced latency and improved throughput compared to single-device deployment [22].

* **T-MAC CPU Renaissance:** Table lookup-based acceleration for low-bit LLM deployment on edge devices, achieving significant performance improvements through CPU-specific optimizations [21].

– **Low-Precision Quantization (High-Definition to Standard Definition):** A critical model compression technique that converts weights and activations in LLMs from high-precision data representations (e.g., 32-bit floating-point, FP32) to low-precision formats (e.g., 8-bit or 4-bit integers, INT4 or INT8) [163]. Similar to converting high-definition photos to standard definition—the file size decreases significantly while preserving essential visual information.
Advanced Quantization Techniques: Recent work highlights quantization's role in edge LLM deployment [25], [110], [164]. Key methods include QLoRA (Quantized Low-Rank Adaptation) for fine-tuning quantized models [165], GPTQ for layer-wise optimization [166], and AWQ for activation-aware quantization [167]. Hardware-aware approaches like HAQ optimize for target devices [129], while mixed-precision strategies (e.g., W4A16) balance accuracy and efficiency [75]. Advanced techniques like AdaRound and BRECQ enable extreme low-bit quantization [168]. Recent work on OmniVLM demonstrates token-compressed sub-billion-parameter vision-language models achieving efficient on-device inference through advanced quantization techniques [169].

– **Knowledge Distillation (KD):** Transfers knowledge from large teacher models to smaller student models, enabling 10-50× size reduction with 90-95% accuracy retention [114]. Figure 10 shows the distillation process where soft targets from teachers guide student training.

* **Advantages:** KD reduces scale and inference cost while preserving most task fidelity.
* **Mechanism:** Softened teacher distributions transfer dark knowledge (class similarity structure).
* **LLM-Specific KD:** KD is crucial for transferring the advanced capabilities of leading proprietary LLMs to more accessible open-source models, and also plays a key role in compressing open-source LLMs for self-improvement. The NVIDIA NeMo framework provides pipelines for LLM pruning and distillation [155].
* **Synergy with Data Augmentation:** Data augmentation significantly enhances LLM performance within KD frameworks by generating contextually rich, skill-specific training data [114].
* **Advanced Distillation Techniques:** Recent work on distilling on-device language models for robot planning demonstrates minimal human intervention approaches, achieving efficient knowledge transfer for specialized edge applications [170].
* **Device-Cloud Collaborative Distillation:** Federated sketching LoRA enables on-device collaborative fine-tuning of LLMs, providing privacy-preserving knowledge distillation across distributed edge devices [71].
* **Multimodal Knowledge Transfer:** Compositional multi-tasking for on-device LLMs leverages distillation to enable efficient task composition and knowledge sharing across different modalities [171].

KD now bridges large proprietary and open student models, enabling capability transfer and iterative self-improvement [114]. Coupled with augmentation, it yields specialized skill gains [114]. Future work should focus on adaptive multi-teacher policies and resource-aware scheduling for heterogeneous edge environments.

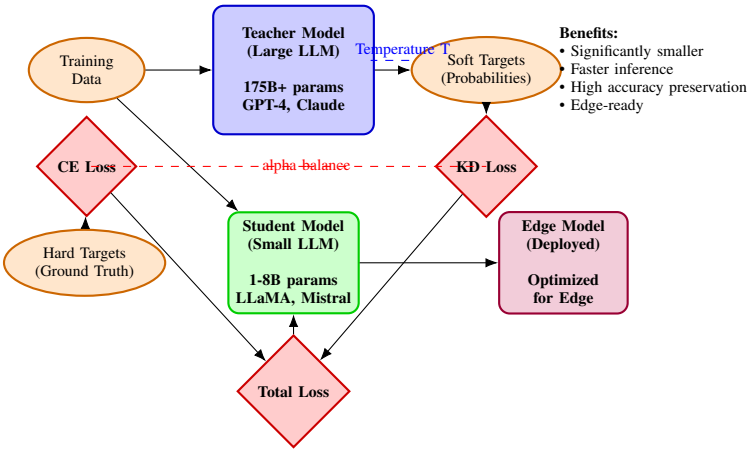


Fig. 10. Knowledge Distillation Architecture for Edge LLM Deployment. The framework illustrates the transfer of knowledge from a large teacher model to a compact student model suitable for edge devices through soft target training, enabling significant model compression while preserving performance.

– **Multimodal Model Optimization:** Vision-language models pose unique optimization challenges due to high-dimensional visual encoders and cross-modal fusion complexity. Recent breakthroughs demonstrate that ultra-lightweight multimodal models can achieve GPT-4V level performance through innovative architectural and optimization strategies [172], [173]:

* **Efficient Visual Encoders:** Apple's FastViTHD demonstrates hybrid CNN-Transformer designs that reduce visual token count while maintaining high-resolution processing capabilities, achieving 3.4× encoder size reduction compared to traditional ViT architectures [78].
* **Multi-Modal Reinforced Training:** MobileCLIP2 employs knowledge transfer from image captioning models and ensemble CLIP encoders to improve small model accuracy without increasing inference cost, storing additional knowledge in reinforced datasets rather than model parameters .

- **Dynamic Resolution Scaling:** FastVLM's architecture supports dynamic image resolution adaptation, allowing models to process high-resolution inputs with minimal computational overhead through intelligent token reduction strategies [78].
- **Cross-Modal Compression:** Techniques like MobileViCLIP extend efficient image-text models to video domains, addressing temporal complexity while maintaining compact model size for mobile deployment [174].
- **Ultra-Lightweight Multimodal Design:** MiniCPM-V 4.0 reports GPT-4V-comparable results on selected evaluations with 4.1B parameters through sparse attention and hierarchical design [70], [175]. Reported efficiency includes sub-2-second first token latency and >17 tokens/second decoding speed on iPhone 16 Pro Max under specified settings.
- **Sparse Long-Context Processing:** InfLLM v2 [176] introduces hierarchical sparse attention that enables efficient processing of ultra-long contexts and cross-modal information fusion, solving the computational bottleneck of traditional attention mechanisms for multimodal inputs. This allows 0.5B parameter models to handle complex multimodal reasoning tasks previously requiring much larger models.
- **Unified Multimodal Architecture:** MiniCPM-V 4.0 integrates text, image, video, and audio processing within a single lightweight framework, achieving comprehensive multimodal understanding through embedded vision and speech encoders [175], [176]. This unified approach eliminates the need for separate specialized models for different modalities.
- **Advanced Quantization and Deployment:** The MiniCPM series supports multiple quantization formats (int4, GGUF) and inference frameworks (llama.cpp, Ollama, vLLM, SGLang), enabling flexible deployment across diverse hardware platforms from mobile devices to edge servers [175], [177], [178].
- **Mobile-Optimized Multimodal Design:** BlueLM-V-3B demonstrates algorithm and system co-design for multimodal LLMs on mobile devices, achieving efficient deployment through hardware-aware optimization and lightweight architectural choices [179], [180].
- **Egocentric Vision-Language Models:** Vinci provides real-time smart assistance through egocentric vision-language models optimized for portable devices, enabling context-aware interaction in wearable computing scenarios [181].

– **Large-Small Model Co-Evolution Architecture:** A revolutionary paradigm that extends beyond traditional knowledge distillation to establish dynamic collaborative systems between large models (LLMs) and small models (SLMs) for resource-constrained scenarios [74]. An overview of this paradigm is shown in Figure 11. Table IV provides a comprehensive overview of representative techniques and their benefits across the co-evolution design space, highlighting how large and small language models can collaborate effectively. This table compares various technical directions including computational load compression, memory optimization, weight quantization, hardware co-design, dynamic collaboration, and evolution efficiency, demonstrating the practical advantages and trade-offs for different edge deployment scenarios.
Table V provides a detailed examination of LLM-specific optimization techniques tailored for edge deployment, covering key methods such as pruning, quantization, knowledge distillation, and architectural innovations. This comprehensive table outlines the mechanisms, edge deployment advantages, and trade-offs for each technique, serving as a practical guide for researchers and practitioners implementing cognitive edge computing solutions.
2) Industrial Case Study: Apple's FastVLM and MobileCLIP2 Edge AI Strategy: Apple's recent open-source release of FastVLM and MobileCLIP2 represents a significant advancement in edge-native vision-language models, demonstrating the practical viability of small model architectures for resource-constrained deployment [66], [78].
FastVLM Architecture and Performance: FastVLM is a multimodal vision-language model optimized for edge devices, featuring a novel hybrid visual encoder called FastViTHD that combines convolutional networks with Transformer architectures. This design enables significant efficiency improvements: 85× faster first-token time-to-first-token (TTFT) compared to similar models like LLaVA-OneVision-0.5B, while reducing visual encoder size by 3.4× [78].
The model achieves this performance through intelligent token reduction strategies that maintain visual fidelity while minimizing computational overhead. FastVLM supports multiple parameter scales (0.5B, 1.5B, 7B) and demonstrates superior performance compared to larger models like Cambrian-1-8B, with 7.9× faster inference while maintaining competitive accuracy across seven vision-language tasks.
MobileCLIP2 Lightweight Design: Complementing FastVLM's speed focus, MobileCLIP2 emphasizes model compactness through multi-modal distillation and data augmentation techniques . The S4 model achieves performance comparable to SigLIP-SO400M/14 on ImageNet-1k while using only half the parameters. On iPhone 12 Pro Max, MobileCLIP2 delivers 2.5× lower latency compared to DFN ViT-L/14, enabling real-time photo search and offline image recognition capabilities [184].
Apple's Dual-Track Edge AI Strategy: Unlike competitors focusing primarily on cloud-based large

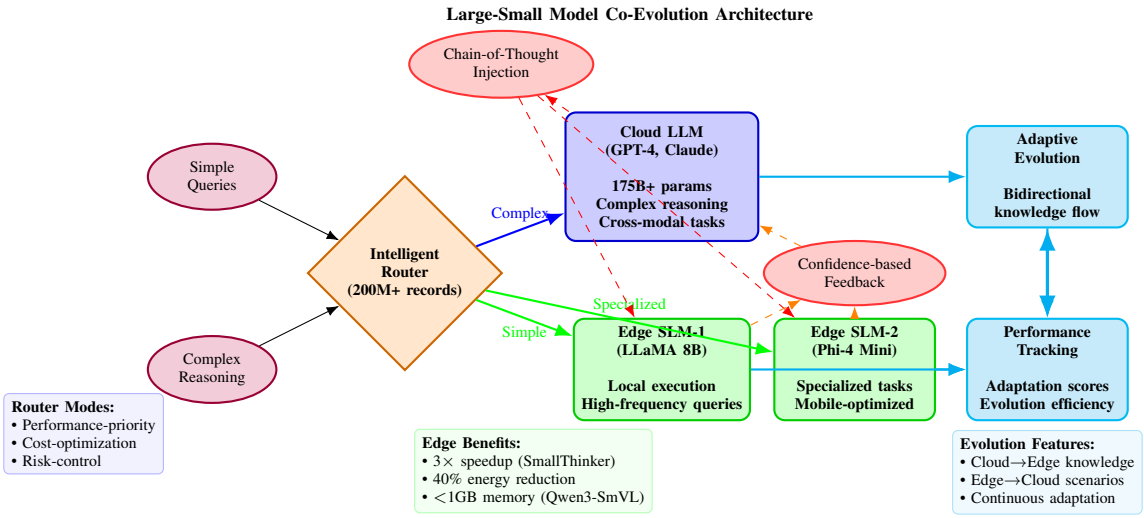


Fig. 11. **Large-Small Model Co-Evolution Architecture:** Dynamic collaborative framework showing knowledge transfer, router-based task allocation, collaborative training mechanisms, and resource optimization strategies for edge deployment [74], [182].

TABLE IV
LARGE–SMALL MODEL CO-EVOLUTION: REPRESENTATIVE TECHNIQUES AND BENEFITS

Technical Direction	Representative Solution	Key Benefit	Scenario	Notes
Computational Load Compression	SmallThinker Two-Level Sparsity	3× speedup, 40% energy reduction	Mobile devices, IoT	Hierarchical sparsity
Memory Optimization	Phi-4-Mini Grouped Query Attention [182]	1/3 memory for long context	Long context tasks	Attention optimization
Weight Quantization	Qwen3-SmVL INT4	700MB model, runs in 1GB VRAM	Android product recognition	Adaptive quantization
Hardware Co-design	WebGPU Local Inference [183]	Browser-native Llama 2	Privacy-sensitive	Web acceleration
Dynamic Collaboration	RouterEval Task Allocation	50ms routing latency	Real-time applications	Intelligent dispatch
Evolution Efficiency	Phi-4-Mini Efficiency [182]	90% large-model performance/parameter	Resource-constrained	Parameter efficiency

language models, Apple has pursued a comprehensive dual-track approach: cloud-scale models for complex tasks and edge-native small models for immediate, privacy-sensitive applications. This strategy addresses fundamental challenges in mobile AI deployment while showcasing practical edge AI applications:

- **Real-time Vision Applications:** FastVLM enables instantaneous camera-based text recognition, live subtitle generation, and real-time image captioning with latency low enough to support accessibility features like screen readers
- **Offline Capabilities:** MobileCLIP2 supports photo album semantic search, camera translation, and image-text retrieval without network connectivity, crucial for privacy-sensitive scenarios
- **Privacy Protection:** Local processing ensures user data never leaves the device, aligning with Apple's privacy-first philosophy while maintaining competitive performance
- **Hardware Integration:** Direct integration with Core ML and Swift Transformers toolchain, leveraging Neural Engine and GPU acceleration on A-series/M-series chips for optimal resource utilization
- **Developer Accessibility:** Open-source models with WebGPU demos accessible through Safari browsers, lowering barriers for developer adoption and experimentation

Performance Validation: Community testing confirms FastVLM's exceptional speed, with users reporting real-time text recognition capabilities that match screen reader speeds and seamless integration with assistive technologies. The models demonstrate consistent accuracy-latency trade-offs across different hardware configurations, validating the viability of edge-optimized multimodal architectures.

Industry Case Studies: Apple's FastVLM achieves 85× faster inference with hybrid CNN-Transformer encoders [78], while MiniCPM-V 4.0 delivers GPT-4V performance with 4.1B parameters on mobile devices [70]. Rockchip's RK3588 demonstrates Chinese semiconductor leadership with 6 TOPS NPU capacity [185].

Competitive Landscape: Hardware-software integration drives edge AI advancement, with ecosystem lock-in strategies and open-source/proprietary

TABLE V
LLM-SPECIFIC MODEL OPTIMIZATION TECHNIQUES FOR EDGE DEPLOYMENT

Technique	LLM-Specific Methods/Examples	Mechanism/Working Principle	Edge Deployment Advantages	Trade-offs/Limitations
Pruning	Depth pruning, width pruning, Deep Compression	Selectively removes unimportant components (weights, neurons, layers)	Smaller model size, faster execution; reduced memory/storage; improved energy efficiency	Potential accuracy loss; optimization computation cost; hardware compatibility constraints
Parameter sharing	Weight clustering, low-rank adaptation (LoRA)	Shares weights across multiple layers or components	Significant parameter reduction; discovers more efficient architectures	Potential accuracy degradation; architecture-specific compatibility
Quantization	PTQ, QAT, QLoRA, GPTQ	Converts weights/activations to low-precision (INT4/INT8) formats	Reduced model size; lower memory/storage; improved energy efficiency	Accuracy loss; difficulty selecting optimal precision; hardware compatibility
Knowledge distillation (KD)	Teacher-student, self-distillation, multi-teacher KD	Transfers knowledge from large teacher to compact student model	Reduced model size/computation while maintaining performance	Possible accuracy drop; domain-dependent effectiveness; requires tuning
Low-rank decomposition	SVD training, micro-factorized convolution	Approximates weight matrices using low-dimensional representations	Reduced memory consumption and computation; faster training	Implementation complexity; requires extensive retraining

tensions shaping market evolution [46], [66], [186]. Chinese companies like Rockchip achieve significant automotive electronics breakthroughs [187].

C. System Optimization

System optimization focuses on software frameworks, hardware accelerators, and distributed strategies to enhance the efficiency of AI workloads on edge devices. Figure 12 illustrates the multi-layer system optimization architecture that integrates these components for optimal LLM and AI agent deployment [49], [55], [110].

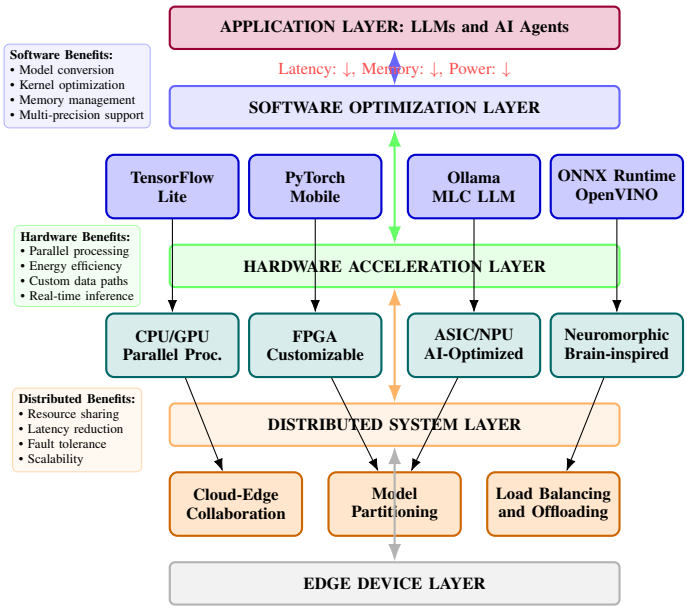


Fig. 12. System Optimization Architecture for Edge AI. The multi-layer framework integrates software frameworks, hardware acceleration, and distributed strategies to optimize LLM and AI agent deployment on resource-constrained edge devices.

1) Practical Deployment Blueprints: We provide four reference blueprints:

- **Smartphone (Thermal-Constrained, 8–16GB RAM):** INT4/INT8 quantization, KV cache eviction policies, speculative decoding with conservative draft length; throttle-aware schedulers; metrics: p50/p90 latency, J/token, thermal stability over 10-min sessions; pitfalls: background app interference and DVFS.
- **Wearable (Ultra-Low Power):** Micro-SLMs with distilled skills, event-driven pipelines, on-demand wake; metrics: idle vs active power, wake-to-first-token; pitfalls: memory fragmentation and sensor interference.
- **Jetson-Class Edge Server:** Mixed-precision (FP16/INT8), tensor-RT kernels, paged KV cache; concurrent sessions in vLLM/SGLang; metrics: throughput vs latency Pareto, energy per query under concurrency; pitfalls: NUMA effects and paging stalls.
- **Base Station/MEC Node:** Partitioned inference with elastic offloading, admission control, privacy zoning; metrics: SLA adherence (p95), offload ratio, e2e energy; pitfalls: network jitter and privacy boundary leaks.

Software Frameworks: Lightweight DL runtimes (TensorFlow Lite, PyTorch Mobile, NCNN, OpenVINO) provide quantization-aware kernels and deployment tooling [6], [164]. LLM serving platforms like Ollama, MLC LLM, and llama.cpp enable efficient edge deployment with MiniCPM-V 4.0 achieving sub-2-second response times [175], [177], [178]. Advanced frameworks like vLLM and SGLang support high-throughput concurrent inference [110], [188], while CPM.cu provides composite acceleration for edge scenarios [176].

Hardware Acceleration: Specialized accelerators are essential for edge LLM deployment. CPUs offer ubiquity but limited parallelism; GPUs provide better processing with higher power consumption; FPGAs enable custom data paths; ASICs and NPUs deliver superior efficiency for AI workloads [33], [34]. Emerging architectures include compute-in-memory (CIM) for

TABLE VI
STANDARDIZED EVALUATION FRAMEWORK FOR EDGE AI WITH LLMS AND AGENTS

Dimension	Latency	Throughput	Energy	Accuracy	Notes
Interactive Inference	p50/p90 response, token/s	concurrent sessions	J/req, J/token	task-specific metrics	On-device vs hybrid
Context Handling	max/avg context, truncation rate	retrieval hit ratio	extra KV traffic	answer consistency	Long-context safety
Privacy/Security	data residency	model leak tests	DP budget/overhead	attack success rate	Threat models
Robustness	OOD error rate	degradation under load	thermal throttling	fail-safe behavior	Graceful fallback
Sustainability	embodied energy	carbon intensity	power caps	SLA compliance	Region-specific mix

Note: Thresholds represent typical requirements for production edge AI systems. Specific applications may require different targets based on use case criticality and resource constraints.

10-100× energy improvements [189], near-memory computing (NMC) for reduced data movement, and transformer-optimized processing units [28], [29], [190]. Recent works demonstrate FPGA-based spatial acceleration achieving 5-20× throughput improvements for LLM inference [26], [27], while NPU-optimized frameworks enable real-time inference on mobile devices [29]–[32]. Hardware-software co-design approaches combine specialized accelerators with optimized kernels to push TOPS/W efficiency toward 100+ for edge multimodal reasoning [27], [33], [34].

Neuromorphic Computing: Brain-inspired computation with spiking neural networks offers exceptional energy efficiency for pattern recognition and real-time sensory processing, promising for low-power edge agents [27], [191], [192]. Neuromorphic systems can achieve substantial energy reductions, often ranging from 100-1000×, for certain cognitive tasks through event-driven processing and sparse activation patterns [191], [192]. Recent advances in neuromorphic computing demonstrate scalable architectures for edge AI applications, with spiking neural networks showing superior efficiency for temporal processing and sensory data analysis [27].

Collaborative Architectures: Model partitioning distributes computation across devices to overcome single-device limitations. Cloud-edge collaboration enables intelligent task offloading, with CE-CoLLM demonstrating 13.81% latency reduction and 84.53% workload offloading [63]. Multi-edge partitioning distributes layers across nearby devices for improved resilience [193]. Dedicated AI co-processors like Rockchip's RK182X enable modular AI enhancement without full system replacement [30]–[32], [194].

Hierarchical Intelligence: Multi-tier architectures distribute intelligence across device-edge-cloud hierarchies, with intelligent routing optimizing for latency, resources, and complexity [26]. Network-aware optimization adapts to connectivity constraints for robust edge-cloud coordination.

A standardized evaluation framework (metrics, methods, and typical targets) is summarized in Table VI.

2) Runtime and KV-Cache Optimization: Edge-side LLM serving performance depends critically on optimizing the decode phase and memory behavior. Practical systems co-design attention kernels, cache layout, and schedulers to increase tokens-per-second while fitting tight memory budgets.

- **Prefill vs. Decode Phases:** *Prefill* (encoder-like pass over the prompt) is bandwidth-bound and benefits from fused attention kernels (e.g., FlashAttention) and operator fusion. *Decode* is latency-critical with batch=1 or small batches; token reuse via KV-cache and speculative decoding dominate throughput.
- **Paged KV-Cache:** vLLM introduces paged attention that manages KV memory in fixed-size pages with a GPU-resident allocator, enabling high reuse, low fragmentation, and stable throughput under variable-length prompts and long sessions [110]. This abstraction decouples sequence length from contiguous allocation, improving latency predictability.
- **Continuous Batching and Scheduling:** Schedulers that merge requests on the fly (a.k.a. continuous batching) and reorder steps across sequences maximize GPU utilization without user-perceived delay. SGLang/vLLM demonstrate large throughput gains under mixed-length, multi-tenant workloads [110], [188].
- **Streaming and KV-Cache Compression:** StreamingLLM-style techniques reduce cache cost by evicting or downscaling low-saliency tokens, head-wise/key-wise sparsification, or low-bit compression of KV tensors with accuracy-aware heuristics. These approaches retain perplexity while shrinking memory footprint for long-context sessions [195].
- **Mobile System-Service (LLMaaS) and Elastic KV:** Treating LLMs as a mobile OS service introduces stateful execution where KV-cache persists across app invocations. Recent systems decouple app and LLM memory management with chunk-wise, tolerance-aware KV compression and lifecycle policies to minimize context-switch latency on devices [196]. Complementary designs elastically adapt model capacity and KV residency for mobile scenarios, trading accuracy for footprint and responsiveness on demand [197].
- **Offloading under Memory Pressure:** FlexGen and related systems split weights/activations/KV across GPU/CPU/NVMe with overlapping prefetch and compute. Careful pipeline design sustains near-GPU-only throughput when memory is insufficient for full-resident deployment [198].

- **Speculative Decoding Compatibility:** Draft-and-verify methods (e.g., MTP/EAGLE/FR-Spec) are complementary to cache optimizations; hierarchical designs co-tune draft length and verification batch to maintain cache locality and reduce verifier stalls [159], [160], [199].
- **Kernel and Memory Co-Design:** Modified FlashAttention with compressed masks, fused rotary-embedding + QKV projection, and register/shared-memory tiling reduce HBM traffic. Practical allocators prioritize: model weights -> intermediate buffers -> KV-cache, with lifetime-aware reuse and NUMA-conscious placement (cf. CPM.cu, vLLM) [110], [111], [176].
- **On-Device Smartphone Scheduling:** Mobile pipelines co-optimize CPU/GPU/NPU scheduling, memory residency, and batching to sustain low-latency decoding within single-digit watt budgets; recent systems demonstrate fast LLM inference directly on smartphones [200].

These runtime strategies combine with quantization (e.g., W4A16 via Marlin/GPTQ), sparse attention, and batching heuristics to deliver stable edge throughput with bounded latency and power draw.

Emerging System Technologies: Recent advances in mixture-of-experts (MoE) routing enable dynamic expert selection for task-adaptive computation, reducing energy by 2-3× for specialized workloads [201]. Neuromorphic computing explores spiking neural networks for energy-efficient temporal processing, achieving 100-1000× energy reductions for certain cognitive tasks [202]. Hardware-software co-design with dedicated AI accelerators (e.g., NPU clusters) pushes TOPS/W efficiency toward 100+, enabling real-time multimodal reasoning on edge devices [27].

V. Applications of Edge AI with LLMs and AI Agents

The convergence of LLMs, AI Agents, and Edge AI enables application patterns across latency-sensitive, privacy-critical, and bandwidth-constrained domains. Reported improvements are context-specific; we cite sources and avoid universal "5–50×" claims.

A. Enhanced Existing Edge AI Applications with Quantitative Impact

LLM-powered edge systems show measurable gains over prior task-specific baselines (energy/latency references: [1], [2], [52], [55]).

- **Healthcare Edge Systems:** On-device medical LLMs support real-time decision support and personalized recommendations while preserving privacy via local processing [37], [43], [203]. EHR-MCP retrieves clinical information from hospital EHRs through MCP for autonomous task execution [204]. FRAME uses a generate–evaluate–reflect loop to improve medical insights [205]. Fleming-R1 targets verifiable medical reasoning for expert-level clinical tasks [206].
- **Autonomous Vehicle Intelligence:** On-device LLMs provide multimodal perception and sub-50ms decisions for safety-critical driving [43], [49]. Representative systems include: TRR Agent, which retrieves and interprets traffic rules via RAG for interpretable decisions across regions [207]; DriVLMe, enabling natural human–vehicle communication and long-horizon navigation [208]; LLM-based misbehavior detection for sign/motion authenticity in C-ITS [209]; and V2V-LLM, which fuses multi-vehicle perception for grounding and planning [210].
- **Industrial IoT and Manufacturing:** Edge LLMs power natural-language interfaces and predictive maintenance; multimodal models analyze sensors in real time [43]. BiGAT-ID attains 99.34% on EdgeIIoTset with 0.0001s inference for real-time intrusion detection [211]. LLM+IR improves defect localization in flexible manufacturing [212]; SCCE outlines foundation-model IIoT across sensing–compute–connectivity–evolution [213]; and DID+RL edge–cloud schemes enhance task offloading and generalization [40].
- **Smart City Infrastructure:** Edge LLMs enable intelligent traffic management systems that process real-time video feeds and sensor data for optimized traffic flow and emergency response [43]. Privacy-preserving natural language interfaces allow citizens to interact with city services through voice commands processed locally on edge devices. LLMs enhance urban computing across transportation, public safety, and environmental monitoring domains, improving data analysis and decision-making capabilities [214]. In urban planning, LLMs serve as intelligent assistants for synthesizing conceptual ideas, generating urban designs, and evaluating planning outcomes through advanced generation and simulation capabilities [215].
- **Agricultural Precision Systems:** On-device LLMs support crop health, pest detection, and irrigation optimization via local natural-language queries; multimodal drone+sensor inputs enable offline advice in rural settings [43]. Domain stacks include AgriGPT with Tri-RAG for grounded reasoning [216], PEZEGO for pest management [217], and AgriBench/MM-LUCAS for evaluation [218]; LLMs also streamline extension services with location-aware guidance [219].
- **Retail and Commercial Applications:** Edge LLMs power private, sub-200ms shopping assistants and analytics [43]. CuSMer combines semi-supervised learning with model merging for robust multimodal intent recognition [220]. Use-case shopping leverages instruction tuning [221]; generative-agent simulations study search vs personalization [222]; and emerging LLM app stores enable mining, risk analysis, and market dynamics [223].

B. Breakthrough On-Device Language Model Implementations

The maturation of on-device language models has produced several landmark implementations that demonstrate practical feasibility and performance breakthroughs across diverse deployment scenarios [43].

- **Google Gemini Nano:** Google's mobile-native multimodal model achieves competitive performance through

4-bit quantization and Tensor Processing Unit integration, enabling offline natural language processing and accessibility features like real-time audio descriptions in Pixel devices [46].

- **Nexa AI Octopus Series:** This breakthrough model exceeds GPT-4 accuracy in function calling tasks with 95% context length reduction, enabling sophisticated agent orchestration and multi-step reasoning directly on edge devices through functional token compression [47].
- **Apple OpenELM:** Apple's systematically scaled models demonstrate remarkable efficiency improvements for iOS ecosystem deployment, providing accessible APIs for developers to incorporate advanced language understanding into mobile applications within Apple's privacy-first framework [66].
- **Microsoft Phi-3:** Microsoft's 3.8B parameter model achieves performance comparable to larger models through innovative training strategies and architectural optimization, enabling cross-platform deployment across mobile devices, embedded systems, and edge servers [45].
- **Assistive Technology Applications:** On-device language models provide immediate, privacy-preserving support for individuals with disabilities, including real-time image-to-text conversion for visual impairment, sign language recognition and translation, and cognitive assistance for memory impairments through personalized AI support [43].

C. Current Deployable Edge AI Applications

Despite technical challenges, several practical edge AI applications have achieved commercial deployment, demonstrating the viability and immediate benefits of on-device intelligent systems. Current AI smartphone development follows three primary technical routes: on-device AI, hybrid on-device-cloud AI with proprietary models, and hybrid on-device-cloud AI with third-party models. On-device AI offers distinct advantages including rapid response times, enhanced privacy protection, and reduced network dependency.

Mobile AI Assistants and Voice Processing:

- **iPhone Siri Integration:** Apple's Neural Engine, introduced with the A11 Bionic in 2017, powers Siri and Apple Intelligence services through on-device processing, achieving 38 TOPS performance in the M4 chip while maintaining energy efficiency and user privacy. Integrated with Core ML framework, it enables real-time voice recognition, natural language understanding, and conversational AI without cloud dependency. Advanced features include multi-language support, contextual awareness, and seamless integration with iOS ecosystem applications for enhanced user experience and accessibility [6].
- **Huawei Celia and HarmonyOS AI:** Huawei's Celia virtual assistant, developed for HarmonyOS and EMUI devices without Google services, supports local voice translation across 50+ languages, real-time object recognition via camera, contextual smart photo organization, and intelligent scene understanding. In China, it evolves into Xiaoyi with PanGu-Σ 3.0 AI model integration on HarmonyOS 4.0, enabling advanced AI capabilities including multimodal interaction, personalized recommendations, and ecosystem-wide device coordination while maintaining on-device processing for privacy and responsiveness [7].
- **Samsung Galaxy AI Features:** Samsung Galaxy AI integrates on-device and cloud processing to support real-time call translation across multiple languages, intelligent photo enhancement with ProVisual Engine, context-aware app recommendations via Gemini Live, productivity tools like Now Brief for content summarization, and advanced camera features for scene optimization. Powered by dedicated NPU acceleration on Exynos platforms, it emphasizes privacy through local data processing while enabling advanced AI capabilities including voice-to-text transcription, smart reply suggestions, and personalized user experiences across supported Galaxy devices [8].

Professional and Development Tools:

- **Local Code Completion:** Integrated development environments like VS Code and JetBrains IDEs incorporate on-device AI models for intelligent code suggestion, completion, and refactoring assistance. These tools provide instant feedback without transmitting proprietary source code to external servers, enabling secure development workflows in air-gapped environments and maintaining intellectual property protection. Advanced features include context-aware code generation, bug detection, and automated testing assistance across multiple programming languages [9].
- **Offline AI Writing Assistants:** Professional writing tools like DeepWriter and Grammarly's offline mode offer comprehensive local grammar correction, style optimization, content enhancement, and readability analysis for sensitive document processing. These applications maintain complete confidentiality in professional environments, supporting legal document review, academic paper editing, and business communication without internet connectivity. Advanced features include tone adjustment, plagiarism detection, and multi-language document processing [224].
- **Local Translation and Transcription:** Enterprise-grade translation platforms and meeting tools support real-time document translation and audio transcription without network dependency, crucial for confidential business negotiations, legal proceedings, and international collaboration. These systems handle multiple languages simultaneously, preserve formatting in complex documents, and provide offline speech-to-text capabilities for secure environments. Advanced implementations include speaker identification, noise filtering, and integration with productivity suites for seamless workflow automation [10].

Advanced Multimodal Applications:

[6] https://en.wikipedia.org/wiki/Neural_Engine

[7] https://en.wikipedia.org/wiki/Celia_(virtual_assistant)

[8] https://en.wikipedia.org/wiki/Galaxy_AI

[9] https://code.visualstudio.com/docs/copilot/overview

[10] https://github.com/openai/whisper

- **Real-time Video Question Answering:** MiniCPM-V 4.0 enables sophisticated video analysis applications including intelligent monitoring systems, interactive online education support, and automated sports commentary generation with rapid response times on mobile devices, supporting seamless multimedia interaction and content understanding [175].
- **Localized Voice Assistants:** MiniCPM-o 2.6 supports bilingual voice interaction with controllable voice characteristics and ultra-low latency processing, enabling privacy-preserving smart home control, personalized assistance, and natural language device management across diverse linguistic environments [225].
- **Professional OCR and Document Processing:** Advanced text recognition capabilities support document digitization, license plate recognition, and industrial inspection applications with high accuracy while operating entirely offline. These systems provide comparable performance to leading cloud-based solutions for text extraction, form processing, and document workflow automation [11].

Open Source and Research Platforms:

- **Ollama Framework[13]:** Provides a user-friendly interface for deploying various open-source LLMs including LLaMA, Mistral, and Phi models on consumer hardware, enabling researchers and developers to experiment with edge AI capabilities through optimized serving configurations, model management tools, and RESTful APIs for seamless integration. The framework supports custom model fine-tuning, privacy-preserving deployment, and cross-platform compatibility to facilitate academic research and development workflows.
- **llama.cpp Optimization[14]:** Offers highly optimized inference implementations for diverse hardware platforms, supporting advanced quantization techniques, memory-efficient execution across CPU, GPU, and specialized accelerators with minimal dependencies. It provides researchers with low-level control over model inference, enables custom architecture support, and includes performance benchmarking tools for edge computing research and optimization.
- **Apple Silicon and Browser Integration:** Apple's MLX framework provides optimized primitives for machine learning on Apple Silicon, enabling efficient deployment of transformer models with native hardware acceleration and unified memory architecture. Web-based frameworks like Transformers.js and WebLLM enable in-browser execution of transformer models, supporting vision-language models for interactive applications without installation requirements, while facilitating multimodal AI research through hardware-software co-design and debugging capabilities [16] [226].

Industrial AIoT and Edge LLM Deployments:

- **Rockchip RK3588 Platform and Hardware Capabilities:** Rockchip's RK3588 SoC features an octa-core CPU, support for LPDDR4/LPDDR4X/LPDDR5 memory, and a high-performance NPU rated at approximately 6 TOPS with mixed-precision support for INT4/INT8/INT16/FP16 operations. The platform offers 8K video decode/encode capabilities, dual HDMI outputs, multiple CSI camera interfaces, and a Mali-G610 GPU. These capabilities enable diverse edge AI workloads including computer vision and neural inference without cloud dependency, particularly in embedded systems with up to 32 GB memory configurations [227]–[229].
- **Performance Characteristics and Limitations:** Deploying LLMs at the edge involves trade-offs between capability and resource constraints. Inference engines like `llama.cpp` show that quantized models can reduce memory footprint while maintaining acceptable accuracy, though with varying performance impact. Smaller quantized LLMs typically achieve higher throughput and lower latency than larger models, while longer contexts impose substantial memory demands. Hardware limitations emerge with larger parameter counts due to RAM, thermal, and power constraints, often requiring cloud fallback when resource boundaries are exceeded [230], [231].
- **Emerging Applications and Ecosystem Impact:** RK3588-powered systems demonstrate commercial viability across automotive, industrial, and smart infrastructure applications, with particular strength in intelligent cockpits where edge inference improves response latency for voice control and driver monitoring under intermittent connectivity. The platform's automotive-grade reliability supports consistent performance across temperature extremes, while its comprehensive SDK accelerates time-to-market. Manufacturing environments benefit from integrated ISP capabilities for real-time quality control and predictive maintenance [187], [232].

D. Novel Applications Driven by Edge LLMs and Agents

The true potential of edge LLMs and AI Agents lies in enabling entirely new classes of applications that leverage local processing capabilities while maintaining privacy and responsiveness. While fully autonomous deployment remains evolving, several promising application areas are emerging:

- **In-vehicle intelligent cockpits:** Edge LLMs enable multimodal interaction in autonomous vehicles, supporting real-time voice control, driver monitoring, and infotainment under intermittent connectivity, integrating sensors with natural language understanding for contextual assistance and emergency response [207], [229].
- **Embedded robotics:** Robots with on-board LLM capability enable sophisticated natural language interaction, dynamic task planning, and environmental awareness in industrial, service, and exploration applications, supporting complex commands and adaptation without cloud dependency [170].
- **Edge-native generative AI for content creation:** Edge devices support real-time content generation (text, image,

[11] https://github.com/getomni-ai/zerox
[13] https://ollama.com/
[14] https://github.com/ggml-org/llama.cpp
[16] https://github.com/huggingface/transformers.js-examples

audio) via lightweight models with quantization, enabling privacy-preserving creative workflows on personal devices with immediate feedback and reduced latency [224].
- **Decentralized collaboration and knowledge reuse:** Federated learning enables edge agents to share knowledge in privacy-preserving ways, creating collective intelligence without centralizing data, supporting collaborative problem-solving across heterogeneous devices [98].
- **Personalized on-device AI assistants:** Edge LLM agents provide deeply personalized, always-on assistance understanding user context and preferences without compromising privacy, offering proactive suggestions and seamless integration across applications [91].
- **Healthcare monitoring and personalized care:** Edge-deployed LLMs enable continuous health monitoring and personalized care recommendations on wearable devices, analyzing vital signs and behavioral patterns while maintaining HIPAA compliance [43].
- **Adaptive educational assistants:** On-device educational LLMs provide personalized learning experiences adapting to individual needs without internet connectivity, offering real-time tutoring and curriculum personalization in remote environments [5].
- **Environmental monitoring and conservation:** Edge AI agents process sensor data for ecological monitoring, detecting changes and coordinating conservation efforts in remote locations without network connectivity [6].
- **Smart infrastructure maintenance:** LLM-powered edge systems enable predictive maintenance and intelligent monitoring of critical infrastructure, analyzing sensor data and providing recommendations autonomously in remote locations [87].

VI. Future Research Directions and Emerging Paradigms

The convergence of LLMs, AI Agents, and Edge computing represents a nascent field with transformative potential requiring breakthrough research across multiple dimensions to achieve ubiquitous intelligent systems [74], [85].

A. Next-Generation Edge AI Architectures

- **Neuromorphic-LLM Hybrid Systems:** Integrating neuromorphic computing principles with LLM architectures may enable order-of-magnitude energy-efficiency gains for event-driven inference [233], [234]. Open challenges include spike-based attention mechanisms, temporal credit assignment for transformer-style models, and hybrid analog–digital pipelines. Aspirational targets (subject to validation) include sub-watt operation for compact models and sub-10ms end-to-end latency under edge constraints. Key research questions include: How can spiking neural networks be effectively integrated with transformer attention mechanisms? What training algorithms can enable temporal credit assignment in large-scale language models?
- **Quantum-Enhanced Edge AI:** Quantum methods for optimization, sampling, or selected linear-algebra subroutines could complement edge AI workflows [235]. Near-term work includes quantum-inspired optimizers for compression and privacy-aware federated learning protocols. Timelines remain uncertain; some roadmaps anticipate task-specific advantages emerging in the 2030s for narrowly scoped problems, pending hardware and algorithmic progress.
- **Brain-Computer Interface Integration:** Tight coupling of neural interfaces with edge LLMs targets ultra-low latency (single-digit milliseconds) pipelines [236]. Priorities include efficient neural signal processing on-device, privacy-preserving analysis, and adaptive personalization. Representative applications include assistive technologies, cognitive augmentation, and rehabilitation; robust clinical validation is required before broad deployment.

B. Towards More Flexible Edge AI

- **Adaptive LLM and Agent Architectures:** Future edge agents will require dynamic architectures that can adapt their complexity and resource consumption based on real-time factors like available power, network conditions, or task priority. This includes dynamic model pruning, adaptive quantization, and on-demand loading of model components. Potential research paths include developing reinforcement learning-based adaptation policies and creating modular architecture frameworks that can reconfigure themselves at runtime [63], [96].
- **Rapid Fine-tuning on Edge:** Developing efficient techniques for rapid, on-device fine-tuning or personalization of LLMs and agent policies will enable faster adaptation to user preferences or new environmental conditions without extensive retraining cycles or cloud dependency [71], [165], [235].

C. Towards More Secure Edge AI

- **Enhanced Privacy-Preserving Techniques:** Beyond current methods, research must focus on integrating more robust privacy techniques like advanced differential privacy [237] and homomorphic encryption more seamlessly into edge LLM and agent workflows, especially for sensitive data processing and federated learning scenarios. Specific research directions include developing lightweight homomorphic encryption schemes optimized for transformer architectures and creating privacy-preserving knowledge distillation protocols for edge environments [98].
- **Robust Agent Behavior and Trust:** Ensuring the trustworthiness and verifiable behavior of autonomous edge agents is paramount. This includes developing mechanisms for auditing agent decisions, detecting malicious or unintended actions, and building resilient systems against adversarial attacks [13], [74].
- **Quantization-Aware Attacks and Robustness:** Low-bit deployment introduces unique threat surfaces. Bit-flip/fault-injection attacks on quantized weights or activations, adversarial rounding, and side-channel leakage from deterministic kernels can degrade reasoning fidelity or induce targeted failures. Robustness strategies include

error-detecting encodings, randomized rounding, per-layer sensitivity hardening, and post-deployment anomaly detection. We recommend reporting robustness under quantization- and fault-specific attacks alongside accuracy/latency/energy metrics [238]–[240].

- **Standards and Compliance Alignment:** Align with NIST AI RMF and ISO/IEC 23894 for risk management and safety practices in on-device AI [241]. Evaluation checklists should include: (i) privacy boundaries (local logging/telemetry policies; on-device vs cloud processing), (ii) safety guardrails (prompt-injection defenses, on-device content filtering), (iii) robustness to quantization/fault/side-channel attacks with task-level impact analysis, and (iv) incident reporting and traceability.
- **ISO/IEC 27001 Principles for Edge AI Security Management:** While specific adaptations are nascent, applying the principles of ISO/IEC 27001 [242] (a systematic approach to managing sensitive company information) to Edge AI deployments can provide a robust framework for identifying risks, implementing controls (e.g., access control, encryption), and continuously improving the security posture of edge ecosystems [243]. This shifts focus from purely technical solutions to comprehensive security management.
- **Blockchain for Secure Data Sharing and Trust:** Blockchain technology can enhance security and privacy in decentralized edge environments by providing immutable ledgers for data provenance, secure multi-party computation for collaborative AI, and verifiable execution of smart contracts for agent coordination [244]. Future work should evaluate its practical integration and scalability for LLM and agent-driven edge applications.

D. Towards More Collaborative Edge AI

- **Edge-Edge and Cloud-Edge Collaboration:** Research needs to refine seamless collaboration frameworks that allow LLMs and agents to dynamically distribute tasks and knowledge across edge devices and cloud resources, optimizing for latency, throughput, and energy efficiency [63], [73], [92].
- **Federated Learning for Heterogeneous Devices:** Addressing the challenges of federated learning in highly heterogeneous edge environments (diverse computational power, varying data distributions, intermittent connectivity) is crucial for collaborative on-device model training [71], [245]. This includes optimizing communication efficiency and aggregation strategies.
- **Multi-Agent Systems (MAS) at the Edge:** Designing and orchestrating complex multi-agent systems where numerous LLM-powered agents collaborate to achieve larger goals requires advanced coordination mechanisms, communication protocols, and conflict resolution strategies optimized for distributed edge deployments [98].
- **Advanced Co-Evolution Architectures:** The next generation of large-small model collaborative systems presents several critical research challenges [246]–[249]:
  - **Neural Architecture Search for Co-Evolution:** Automated design of optimal LLM-SLM hybrid architectures using advanced NAS techniques, extending beyond traditional single-model optimization to multi-model system design. This includes exploring modular architectures like Qwen3-SmVL's component-based assembly for dynamic capability scaling [149].
  - **Routing Latency Optimization:** Current Router-based scheduling systems introduce approximately 50ms latency overhead, which constrains real-time applications. Future research must focus on ultra-low latency routing algorithms, potentially using specialized hardware acceleration or predictive scheduling based on task pattern analysis [24].
  - **Security in Multi-Model Systems:** Co-evolution architectures expand attack surfaces through increased model interactions and data flows. Research priorities include adversarial robustness across model boundaries, secure knowledge transfer protocols, and distributed intrusion detection systems specifically designed for collaborative AI environments [74].
  - **Carbon-Efficient Co-Evolution:** Developing green scheduling strategies that optimize task allocation based on energy source availability and carbon footprint, prioritizing renewable energy-powered edge nodes and implementing dynamic power management across the collaborative network [28], [250].

E. Towards Resource-Aware Co-Evolution Systems

- **Dynamic Adaptation Mechanisms:** Future systems must implement sophisticated resource monitoring and adaptation strategies that can dynamically adjust the collaboration ratio between large and small models based on real-time constraints including battery level, thermal conditions, network bandwidth, and computational load [74].
- **Edge-Native Evolution Metrics:** Developing specialized evaluation frameworks for co-evolution systems that consider edge-specific factors such as adaptation speed (cold-start performance on new tasks), evolution efficiency (parameter update-to-performance ratio), energy consumption per knowledge transfer, and system resilience under resource constraints [251].
- **Contextual Knowledge Transfer:** Advancing beyond static knowledge distillation to context-aware, bidirectional knowledge flows where small models not only receive guidance from large models but also contribute domain-specific insights, local environmental adaptations, and user behavior patterns back to the collaborative system [182], [183].

F. Towards More Efficient Edge AI

- **Continued Algorithm and Hardware Improvements:** Ongoing research into novel compression techniques, more efficient LLM architectures, and next-generation AI accelerators (including those co-designed with software

stacks) will continue to push the boundaries of what is possible on edge devices [179].

- **Resource-Aware Agent Decision-Making:** Future agents should possess an inherent awareness of their own and surrounding devices' computational, memory, and energy resources, enabling them to dynamically adapt their behavior or offload tasks to optimize overall system efficiency [24].

G. Usability and Performance Evaluation

- **Advanced Framework Performance Benchmarking:** Comprehensive evaluation of next-generation edge inference frameworks demonstrates significant acceleration potential through composite optimization strategies [176], [252]:
  CPM.cu Framework Performance Analysis: Extensive benchmarking of the CPM.cu lightweight inference framework reveals substantial performance improvements across diverse deployment scenarios. Testing with MiniCPM 4.0-8B model demonstrates consistent 5× acceleration in regular inference scenarios compared to baseline implementations [176]. Under memory-constrained conditions, the framework achieves exceptional performance gains of up to 220× speedup through intelligent memory management and composite acceleration integration.
  Speculative Sampling Acceleration Metrics: FR-Spec frequency-ranked speculative sampling integration achieves measurable performance improvements with 75% reduction in LM Head computational overhead and 1.12× average speedup over EAGLE-2 baseline [159]. When combined with CPM.cu's sparse attention mechanisms, compound acceleration effects reach 1.8× to 2.3× improvement in token generation throughput depending on model size and attention pattern complexity.
  Quantization Performance Impact Assessment: GPTQ integration with Marlin format conversion maintains model accuracy while achieving 3-4× memory reduction and corresponding inference acceleration. W4A16 mixed-precision quantization demonstrates optimal balance between memory efficiency and computational precision, enabling deployment of 8B parameter models on devices with 4GB memory constraints [253].
- **Standardized Usability Frameworks:** To assess the practical efficiency and effectiveness of Edge AI deployments, integrating standardized usability evaluation metrics is essential. The ISO 9126 usability framework [254], for instance, provides a structured model encompassing aspects like understandability, learnability, operability, and attractiveness. Applying these metrics to Edge AI could involve:
  - **Understandability (Ease of learning and user guidance):** How easy is it for developers and data scientists to deploy Edge AI models? Measuring the time required for first-time deployment.
  - **Learnability (Ease of adoption):** How quickly can users understand Edge AI outputs? Assessing how well explainable AI (XAI) techniques enhance model interpretability and conducting user tests with domain experts (e.g., healthcare professionals) to measure the time-to-understanding of AI decisions.
  - **Operability (Ease of operation):** How reliably do edge LLMs/agents perform their tasks in real-world scenarios?

  This will provide a more holistic evaluation beyond just technical benchmarks.
- **Standardized Evaluation Framework:** Table VI provides a comprehensive framework summarizing key dimensions and representative metrics for evaluating edge AI systems with LLMs and agents.
- **Safety/Compliance Dimension:** Extend the evaluation framework with a Safety/Compliance column capturing privacy boundaries (on-device vs cloud processing), guardrails (prompt injection defenses, content filters), robustness under quantization/fault/side-channel threats, and incident traceability, aligned with NIST AI RMF and ISO/IEC 23894 [255].
  a) Standardized Measurement Protocol (Recommended).: For cross-paper comparability, report for latency/throughput/energy/accuracy:
  - Workload: prompt length, generation length, temperature/top-k/p, beam size; task type and dataset split.
  - Precision and Compression: numeric precision (INT4/8/FP16), sparsity ratio, KV cache policy, quantization-aware vs post-training.
  - Batching and Concurrency: batch size, concurrent sessions, scheduler policy (FCFS, preemption), speculative decoding settings.
  - Hardware and Thermal: device model, CPU/GPU/NPU clocks, cooling state, thermal throttling policy.
  - Energy Measurement: meter type and scope (device-only vs system vs facility), sampling rate, J/token and Wh/query definitions.
  - Reproducibility: code commit/version, benchmark version/date, configuration files, and seed settings.

H. Research Gaps and Future Work

- **Unified Benchmarking for Edge LLMs and Agents:** The heterogeneity of edge hardware and the diversity of LLM and agent tasks necessitate standardized benchmarks that accurately reflect real-world performance, latency, and energy consumption under various constraints. Research priorities include developing modality-aware reasoning benchmarks that incorporate edge-specific factors like thermal constraints and intermittent connectivity, as well as creating open-source benchmark suites with reproducible energy measurement protocols.
- **Continual Learning and On-Device Adaptation for LLMs:** A major challenge lies in enabling SLMs and other edge-deployed LLMs to continuously learn and adapt from new local data, achieving adaptive fine-tuning capabilities [5].

- **Usability and Performance Evaluation:** Beyond technical metrics, apply ISO 9126 usability criteria—understandability, learnability, operability—to assess deployment efficiency and user comprehension, aided by XAI [6].
- **Model Partitioning:** Partition models across devices to meet accuracy/latency constraints [256]. Adaptive frameworks (e.g., AMP4EC) monitor resources to dynamically partition and schedule, reducing latency and improving throughput [257].
- **Emerging Computing Paradigms:**
  - **Neuromorphic Computing:** This computing paradigm mimics the structure and function of the human brain and promises to surpass traditional computers in energy efficiency and performance [191]. It simulates the working principles of biological neurons through Spiking Neural Networks (SNNs), significantly reducing power consumption and being very suitable for edge AI applications such as IoT devices and sensors [258]. Neuromorphic chips can perform computations directly in memory and run various AI applications with a fraction of the energy consumption of traditional AI platforms [191].
  - **TinyML:** Focuses on running machine learning models on extremely low-power (milliwatt or even microwatt level) microcontrollers. It highly aligns with the energy efficiency goals of edge AI, providing solutions for AI deployment on ultra-low-power devices [5].
- **Explainability (XAI) at the Edge:** Ensuring that LLMs and autonomous agents operating at the edge can provide interpretable explanations for their decisions is crucial for building trust, debugging, and compliance, especially in sensitive applications. Research is needed on resource-efficient XAI techniques suitable for edge devices.
- **Ethical Considerations of Autonomous Edge Agents:** As agents become more autonomous and pervasive, addressing ethical concerns related to bias, fairness, accountability, and the potential societal impact of their decisions is paramount.

Key research questions to prioritize include: How can we develop modality-aware reasoning benchmarks that account for edge-specific constraints? What are the most effective strategies for transparent and reproducible energy reporting in edge deployments? How should safety and alignment evaluation be adapted for resource-constrained edge environments? What testbeds are needed for reproducible multi-agent edge scenarios?

VII. Threats to Validity and Evidence Qualification

Key validity threats include:

- **Data Source Variability:** Reported latency, energy, and accuracy ranges originate from heterogeneous hardware testbeds, batch sizes, precisions, and workload mixes. Cross-paper comparisons risk inconsistency without standardized benchmarking.
- **Hardware / Firmware Revisions:** Accelerator driver or firmware updates can shift measured throughput/efficiency (occasionally by double-digit percentages) invalidating earlier point estimates.
- **Undisclosed Proprietary Model Details:** Parameter counts, training corpora composition, and inference stack optimizations for closed models (e.g., GPT-4) remain partly opaque; extrapolations are avoided or explicitly qualified.
- **Synthetic or Proxy Benchmarks:** Some compression / co-evolution claims rely on limited proxy tasks (e.g., math word problems, selective tool-use scenarios) and may not generalize to broader reasoning or safety-critical settings.
- **Publication Bias:** Positive results (higher compression with minimal loss) are over-represented; negative or neutral findings (e.g., diminishing returns after compound optimizations) less frequently appear, skewing perceived average gains.
- **Energy Measurement Inconsistency:** Studies differ in whether they report wall-plug, SoC package, or accelerator-only power and in stabilization window (steady state vs. burst); energy-per-token normalizations are seldom uniform.
- **Security Metric Ambiguity:** Attack success rates and defense overheads depend strongly on threat model (white vs. black box, adaptive vs. static); generalized percentages risk misinterpretation without scenario framing.
- **Temporal Obsolescence:** Rapid model/hardware iteration can obsolete quantitative ranges within months; we encourage timestamped benchmarking and versioned artifact releases.

A. Future Development Trends and Ecosystem Evolution

The trajectory of edge AI development suggests several transformative trends that will reshape the computational landscape and industrial ecosystem:

Edge-Cloud Collaborative Intelligence: Future AI systems will not replace cloud computing but establish sophisticated collaboration frameworks where edge devices handle immediate, context-sensitive tasks while seamlessly orchestrating with cloud resources for complex reasoning. This hybrid approach will enable graceful degradation under network constraints while maximizing computational efficiency across the entire hierarchy [63], [251].

Ecosystem Convergence and Standardization: The current fragmentation across hardware platforms, software frameworks, and deployment tools will gradually consolidate around interoperable standards. Industry initiatives toward common APIs, model interchange formats, and performance benchmarks will reduce development complexity and accelerate adoption, similar to the evolution of web standards that enabled universal internet applications [6].

Hardware-Software Co-Evolution: The distinction between hardware capabilities and software optimization will blur as specialized AI accelerators (NPUs, neuromorphic chips)

become deeply integrated with model architectures. Custom instruction sets, memory hierarchies, and on-chip learning capabilities will be co-designed with AI model architectures, fundamentally changing how we conceptualize and optimize edge AI systems [5].

Democratization Through Simplification: Just as mobile app development became accessible to millions of developers through simplified frameworks and tools, edge AI deployment will become increasingly democratized. No-code AI platforms, automated optimization tools, and plug-and-play edge AI modules will enable widespread adoption beyond specialist researchers and engineers.

Privacy-Preserving Intelligence Networks: Edge AI will enable new paradigms of distributed intelligence where devices collaborate and learn from each other while preserving individual privacy. Federated learning, differential privacy, and secure multi-party computation will mature into practical frameworks that balance collective intelligence benefits with privacy protection requirements [98].

B. Breakthrough Technical Innovations and Their Implications

Recent advances exemplified by MiniCPM-V 4.0 and related ultra-lightweight multimodal models represent fundamental paradigm shifts that will define the next generation of edge AI systems:

Ultra-Lightweight Architectural Design: The breakthrough achievement of GPT-4V level performance in only 4.1B parameters through MiniCPM-V 4.0 demonstrates that architectural innovation can overcome scale limitations [70], [175]. This validates the principle that specialized, efficiently-designed models can outperform larger general-purpose models in resource-constrained scenarios, potentially revolutionizing mobile AI deployment strategies.

Sparse Long-Context Processing: InfLLM v2's hierarchical sparse attention mechanisms enable ultra-lightweight models to handle complex long-context reasoning tasks previously requiring orders of magnitude more parameters. This innovation fundamentally changes the memory-performance trade-off curve for edge deployment and enables sophisticated document analysis, multi-turn conversations, and contextual reasoning on mobile devices [176], [259].

Comprehensive Multimodal Integration: The unification of text, image, video, and audio processing within single lightweight frameworks eliminates the traditional requirement for separate specialized models [175]. This architectural convergence reduces deployment complexity, memory footprint, and inference latency while enabling richer user interactions and more sophisticated edge applications.

Advanced Deployment Ecosystems: The maturation of deployment frameworks supporting multiple quantization formats (int4, GGUF), inference engines (llama.cpp, Ollama, vLLM, SGLang), and specialized accelerators (CPM.cu) creates unprecedented flexibility for edge AI deployment. This ecosystem development reduces barriers to adoption and enables efficient utilization of diverse hardware platforms.

Mobile-First Performance Optimization: Achieving sub-2-second first token latency and >17 tokens/second sustained performance on consumer smartphones without thermal throttling establishes new benchmarks for mobile AI capabilities [175]. This performance level enables real-time interactive applications previously feasible only on high-end desktop hardware.

Open-Source Innovation Leadership: Open-source efforts such as MiniCPM-V show that collaborative development can outpace proprietary approaches and democratize advanced capabilities [70], [175].

Collectively, these advances move edge AI from experimental prototypes to deployment-ready technology with broad impact across mobile computing, autonomy, and HCI.

VIII. Conclusion

Edge deployment of reasoning-capable models shifts optimization from opportunistic efficiency to a prerequisite for feasibility. Our synthesis shows: (i) single-lever techniques (quantization, pruning, distillation) each deliver distinct early gains but compound stacking exhibits diminishing returns; (ii) architecture co-design (edge-native SLMs, efficient attention variants) increasingly replaces pure post-hoc compression; (iii) adaptive routing and co-evolution frameworks redefine inference as a dynamic decision process rather than a monolithic forward pass; (iv) system and scheduling layers (partitioning, heterogeneous acceleration, memory paging) materially influence end-to-end cognitive latency and should be evaluated jointly with model metrics. This comprehensive framework, as illustrated in Figure 1, integrates cognitive challenges, optimization strategies, applications, and evaluation protocols to preserve reasoning fidelity under stringent resource constraints.

The industrial landscape demonstrates that edge AI is transitioning from experimental research to commercial reality, with major technology companies, chip manufacturers, and AI developers pursuing complementary yet competitive strategies. Current deployable applications in mobile assistants, professional tools, and open-source frameworks validate the technical feasibility while revealing practical constraints and optimization opportunities. The market dynamics underscore the convergence of LLMs, AI agents, and edge computing as a transformative paradigm for ubiquitous intelligence.

Persistent gaps include standardized, modality-aware reasoning fidelity benchmarks, transparent energy / power reporting protocols, robust edge-oriented safety and alignment evaluation, and reproducible multi-agent task suites. Progress will depend on principled cross-layer evaluation artifacts rather than isolated point optimizations. The future of edge AI lies not in replacing cloud computing but in establishing sophisticated collaboration frameworks that leverage the strengths of both paradigms while addressing the fundamental challenges of network dependency, privacy concerns, and personalization limitations that currently constrain AI deployment. Looking ahead, the convergence of LLMs, AI agents, and edge computing promises to democratize access to advanced AI capabilities, enabling intelligent systems that operate seamlessly across diverse environments. As we move toward ubiquitous cognitive edge computing, the focus must shift from mere technical feasibility to ensuring these systems are trustworthy, sustainable, and beneficial to humanity.

The technological trajectory suggests that edge AI will evolve from a specialized optimization challenge to a fundamental computing paradigm, enabling ubiquitous intelligent systems that enhance human capability while preserving privacy and autonomy.

ACKNOWLEDGMENTS

This work was supported by the Institute of Artificial Intelligence and Future Networks, Beijing Normal University. Part of this work was completed during the first author's visiting research at The Hong Kong Polytechnic University. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

Use of AI-Generated Content: In accordance with IEEE guidelines on the use of artificial intelligence (AI)-generated text, the authors disclose that AI-assisted tools were used in limited capacity during the preparation of this manuscript for literature organization, grammar checking, and initial draft structuring. All technical content, analysis, critical insights, evaluations, and conclusions presented in this survey are original work by the authors. Any AI-generated text has been thoroughly reviewed, verified, and substantially edited by the authors to ensure accuracy, originality, and alignment with the authors' expertise and perspectives.

REFERENCES

[1] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[2] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457–7469, 2020.

[3] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge ai: On-demand accelerating deep neural network inference via edge computing," *IEEE transactions on wireless communications*, vol. 19, no. 1, pp. 447–457, 2019.

[4] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[5] X. Wang, Z. Tang, J. Guo, T. Meng, C. Wang, T. Wang, and W. Jia, "Empowering edge intelligence: A comprehensive survey on on-device ai models," *ACM Computing Surveys*, vol. 57, no. 9, pp. 1–39, 2025.

[6] X. Wang and W. Jia, "Optimizing edge ai: A comprehensive survey on data, model, and system strategies," *arXiv preprint arXiv:2501.03265*, 2025.

[7] 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*, vol. 33, pp. 1877–1901, 2020.

[8] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler *et al.*, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022.

[9] 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*, vol. 30, 2017.

[10] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[11] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1 incentivizes reasoning in llms through reinforcement learning," *Nature*, vol. 645, no. 8081, pp. 633–638, 2025. [Online]. Available: https://doi.org/10.1038/s41586-025-09422-z

[12] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.

[13] S. Han, Q. Zhang, Y. Yao, W. Jin, and Z. Xu, "Llm multi-agent systems: Challenges and open problems," *arXiv preprint arXiv:2402.03578*, 2024.

[14] B. Yan, Z. Zhou, L. Zhang, L. Zhang, Z. Zhou, D. Miao, Z. Li, C. Li, and X. Zhang, "Beyond self-talk: A communication-centric survey of llm-based multi-agent systems," *arXiv preprint arXiv:2502.14321*, 2025.

[15] R. Qin, J. Xia, Z. Jia, M. Jiang, A. Abbasi, P. Zhou, J. Hu, and Y. Shi, "Enabling on-device large language model personalization with self-supervised data selection and synthesis," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, 2024, pp. 1–6.

[16] K. Alizadeh, S. I. Mirzadeh, D. Belenko, S. Khatamifard, M. Cho, C. C. Del Mundo, M. Rastegari, and M. Farajtabar, "Llm in a flash: Efficient large language model inference with limited memory," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 12 562–12 584.

[17] Y. Jeon, C. Lee, K. Park, and H.-y. Kim, "A frustratingly easy post-training quantization scheme for llms," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 14 446–14 461.

[18] A. Edalati, M. Tahaei, A. Rashid, V. Nia, J. Clark, and M. Rezagholizadeh, "Kronecker decomposition for gpt compression," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 219–226.

[19] H. Bai, L. Hou, L. Shang, X. Jiang, I. King, and M. R. Lyu, "Towards efficient post-training quantization of pre-trained language models," *Advances in neural information processing systems*, vol. 35, pp. 1405–1418, 2022.

[20] Z. Guan, H. Huang, Y. Su, H. Huang, N. Wong, and H. Yu, "Aptq: Attention-aware post-training mixed-precision quantization for large language models," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, 2024, pp. 1–6.

[21] J. Wei, S. Cao, T. Cao, L. Ma, L. Wang, Y. Zhang, and M. Yang, "T-mac: Cpu renaissance via table lookup for low-bit llm deployment on edge," in *Proceedings of the Twentieth European Conference on Computer Systems*, 2025, pp. 278–292.

[22] M. Zhang, X. Shen, J. Cao, Z. Cui, and S. Jiang, "Edgeshard: Efficient llm inference via collaborative edge computing," *IEEE Internet of Things Journal*, 2024.

[23] R. Kong, Y. Li, Q. Feng, W. Wang, X. Ye, Y. Ouyang, L. Kong, and Y. Liu, "Swapmoe: Serving off-the-shelf moe-based large language models with tunable memory budget," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 6710–6720.

[24] J. Yang, Q. Wu, Z. Feng, Z. Zhou, D. Guo, and X. Chen, "Quality-of-service aware llm routing for edge computing with multiple experts," *IEEE Transactions on Mobile Computing*, no. 99, pp. 1–15, 2025.

[25] G. Qu, Q. Chen, W. Wei, Z. Lin, X. Chen, and K. Huang, "Mobile edge intelligence for large language models: A contemporary survey," *IEEE Communications Surveys & Tutorials*, 2025.

[26] H. Chen, J. Zhang, Y. Du, S. Xiang, Z. Yue, N. Zhang, Y. Cai, and Z. Zhang, "Understanding the potential of fpga-based spatial acceleration for large language model inference," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 18, no. 1, pp. 1–29, 2024.

[27] J. Li, T. Li, G. Shen, D. Zhao, Q. Zhang, and Y. Zeng, "Pushing up to the limit of memory bandwidth and capacity utilization for efficient llm decoding on embedded fpga," in *2025 Design, Automation & Test in Europe Conference (DATE)*. IEEE, 2025, pp. 1–7.

[28] H. Xu, X. Wang, and S. Ji, "Towards energy-efficient llama2 architecture on embedded fpgas," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 5570–5571.

[29] D. Xu, H. Zhang, L. Yang, R. Liu, G. Huang, M. Xu, and X. Liu, "Fast on-device llm inference with npus," in *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, 2025, pp. 445–462.

[30] S. H. Seo, J. Kim, D. Lee, S. Yoo, S. Moon, Y. Park, and J. W. Lee, "Facil: Flexible dram address mapping for soc-pim cooperative on-device llm inference," in *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2025, pp. 1720–1733.

[31] H. Lee, D. Baek, J. Son, J. Choi, K. Moon, and M. Jang, "Paise: Pim-accelerated inference scheduling engine for transformer-based llm," in *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2025, pp. 1707–1719.

[32] W. Sun, M. Gao, Z. Li, A. Zhang, I. Y. Chou, J. Zhu, S. Wei, and L. Liu, "Lincoln: Real-time 50˜100b llm inference on consumer devices with lpddr-interfaced, compute-enabled flash memory," in *2025 IEEE*

International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2025, pp. 1734–1750.

[33] F. Liu, N. Yang, Z. Wang, X. Zhu, H. Yao, X. Xiong, Q. Sun, and L. Jiang, "Ops: Outlier-aware precision-slice framework for llm acceleration," in *2025 Design, Automation & Test in Europe Conference (DATE)*. IEEE, 2025, pp. 1–2.

[34] Z. Yu, S. Liang, T. Ma, Y. Cai, Z. Nan, D. Huang, X. Song, Y. Hao, J. Zhang, T. Zhi *et al.*, "Cambricon-llm: A chiplet-based hybrid architecture for on-device inference of 70b llm," in *2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2024, pp. 1474–1488.

[35] X. Ma, L. Luo, and Q. Zeng, "From one thousand pages of specification to unveiling hidden bugs: Large language model assisted fuzzing of matter $\{IoT\}$ devices," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 4783–4800.

[36] Y. Oliinyk, M. Scott, R. Tsang, C. Fang, H. Homayoun *et al.*, "Fuzzing $\{BusyBox\}$: Leveraging $\{LLM\}$ and crash reuse for embedded bug unearthing," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 883–900.

[37] S. Gilbert, H. Harvey, T. Melvin, E. Vollebregt, and P. Wicks, "Large language model ai chatbots require approval as medical devices," *Nature Medicine*, vol. 29, no. 10, pp. 2396–2398, 2023.

[38] Y. Yao, T. Yu, A. Zhang, C. Wang, J. Cui, H. Zhu, T. Cai, C. Chen, H. Li, W. Zhao *et al.*, "Efficient gpt-4v level multimodal large language model for deployment on edge devices," *Nature Communications*, vol. 16, no. 1, p. 5509, 2025.

[39] Z. Hu, M. Kemertas, L. Xiao, C. Phillips, I. Mohomed, and A. Fazly, "Realizing efficient on-device language-based image retrieval," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 9, pp. 1–18, 2024.

[40] Y. Ren, H. Zhang, F. R. Yu, W. Li, P. Zhao, and Y. He, "Industrial internet of things with large language models (llms): an intelligence-based reinforcement learning approach," *IEEE Transactions on Mobile Computing*, 2024.

[41] Y. Zhang, H. Wang, Q. Bai, H. Liang, P. Zhu, G.-M. Muntean, and Q. Li, "Vavlm: Toward efficient edge-cloud video analytics with vision-language models," *IEEE Transactions on Broadcasting*, 2025.

[42] X. Sun, J. Li, and Q. Li, "Research on security situation analysis and intelligent disposal technology of edge side area," in *2020 International Conference on Data Processing Techniques and Applications for Cyber-Physical Systems: DPTA 2020*. Springer, 2021, pp. 1563–1567.

[43] D. Xu, W. Yin, H. Zhang, X. Jin, Y. Zhang, S. Wei, M. Xu, and X. Liu, "Edgellm: Fast on-device llm inference with speculative decoding," *IEEE Transactions on Mobile Computing*, 2024.

[44] 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.

[45] M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison, R. J. Hewett, M. Javaheripi, P. Kauffmann *et al.*, "Phi-4 technical report," *arXiv preprint arXiv:2412.08905*, 2024.

[46] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[47] W. Chen and Z. Li, "Octopus v2: On-device language model for super agent," *arXiv preprint arXiv:2404.01744*, 2024.

[48] W. Shi, G. Pallis, and Z. Xu, "Edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1474–1481, 2019.

[49] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A survey on edge computing systems and tools," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1537–1562, 2019.

[50] Y. Zheng, Y. Chen, B. Qian, X. Shi, Y. Shu, and J. Chen, "A review on edge large language models: Design, execution, and applications," *ACM Computing Surveys*, vol. 57, no. 8, pp. 1–35, 2025.

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

[52] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[53] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE communications surveys & tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[54] G. Muhammad and M. S. Hossain, "Emotion recognition for cognitive edge computing using deep learning," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 16 894–16 901, 2021.

[55] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," in *Low-power computer vision*. Chapman and Hall/CRC, 2022, pp. 291–326.

[56] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 09, 2020, pp. 13 693–13 696.

[57] N. Corporation, "Nvidia h100 sxm5 gpu technical specifications," 2024, official specifications: 80GB HBM3, 3.35TB/s memory bandwidth, 1979 TOPS (Sparse INT8), 700W TDP. [Online]. Available: https://www.nvidia.com/en-us/data-center/h100/

[58] ——, "Nvidia jetson agx orin developer kit product specifications," 2024, technical specifications: 64GB unified memory, 204.8GB/s bandwidth, 275 TOPS AI performance, 60W max power. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/

[59] A. Inc. and AnandTech, "Apple a17 pro soc technical analysis and performance benchmarks," 2024, neural Engine: 35 TOPS; System memory: 8GB; Estimated memory bandwidth: 68GB/s; Process: TSMC N3B 3nm. [Online]. Available: https://en.wikipedia.org/wiki/Apple_A17

[60] X. Wang and W. Jia, "A feature weighting particle swarm optimization method to identify biomarker genes," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2022, pp. 830–834.

[61] X. Wang, Y. Wang, Z. Ma, K.-C. Wong, and X. Li, "Exhaustive exploitation of nature-inspired computation for cancer screening in an ensemble manner," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 21, no. 5, pp. 1366–1379, 2024.

[62] C. Tian, X. Qin, K. Tam, L. Li, Z. Wang, Y. Zhao, M. Zhang, and C. Xu, "Clone: Customizing llms for efficient latency-aware inference at the edge," *arXiv preprint arXiv:2506.02847*, 2025.

[63] H. Jin and Y. Wu, "Ce-collm: Efficient and adaptive large language models through cloud-edge collaboration," *arXiv preprint arXiv:2411.02829*, 2024.

[64] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia *et al.*, "Glm-130b: An open bilingual pre-trained model," in *The Eleventh International Conference on Learning Representations*, 2023.

[65] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[66] S. Mehta, M. H. Sekhavat, Q. Cao, M. Horton, Y. Jin, C. Sun, S. I. Mirzadeh, M. Najibi, D. Belenko, P. Zatloukal *et al.*, "Openelm: An efficient language model family with open training and inference framework," in *Workshop on Efficient Systems for Foundation Models II@ ICML2024*, 2024.

[67] D. Xu, W. Yin, X. Jin, Y. Zhang, S. Wei, M. Xu, and X. Liu, "Llmcad: Fast and scalable on-device large language model inference," *arXiv preprint arXiv:2309.04255*, 2023.

[68] Z. Liu, C. Zhao, F. Iandola, C. Lai, Y. Tian, I. Fedorov, Y. Xiong, E. Chang, Y. Shi, R. Krishnamoorthi *et al.*, "Mobilellm: Optimizing sub-billion parameter language models for on-device use cases," in *Forty-first International Conference on Machine Learning*, 2024.

[69] T. Çöplü, M. Loedi, A. Bendiken, M. Makohin, J. J. Bouw, and S. Cobb, "A performance evaluation of a quantized large language model on various smartphones," *arXiv preprint arXiv:2312.12472*, 2023.

[70] Y. Yao, T. Yu, A. Zhang, C. Wang, J. Cui, H. Zhu, T. Cai, C. Chen, H. Li, W. Zhao *et al.*, "Efficient gpt-4v level multimodal large language model for deployment on edge devices," *Nature Communications*, vol. 16, no. 1, p. 5509, 2025.

[71] W. Fang, D.-J. Han, L. Yuan, S. Hosseinalipour, and C. G. Brinton, "Federated sketching lora: On-device collaborative fine-tuning of large language models," *arXiv preprint arXiv:2501.19389*, 2025.

[72] X. Li, D. Spatharakis, S. Ghafouri, J. Fan, H. Vandierendonck, D. John, B. Ji, and D. Nikolopoulos, "Sled: A speculative llm decoding framework for efficient edge serving," *arXiv preprint arXiv:2506.09397*, 2025.

[73] T. Sun, P. Wang, and F. Lai, "Disco: Device-server collaborative llm-based text streaming services," *arXiv preprint arXiv:2502.11417*, 2025.

[74] H.-a. Gao, J. Geng, W. Hua, M. Hu, X. Juan, H. Liu, S. Liu, J. Qiu, X. Qi, Y. Wu *et al.*, "A survey of self-evolving agents: On path to artificial super intelligence," *arXiv preprint arXiv:2507.21046*, 2025.

[75] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, "A white paper on neural network quantization," *arXiv preprint arXiv:2106.08295*, 2021.

[76] 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.

[77] X. Wang, J. Wu, Y. Yichen, D. Cai, M. Li, and W. Jia, "Demonstration selection for in-context learning via reinforcement learning," in *Forty-second International Conference on Machine Learning*, 2025.

[78] P. K. A. Vasu, F. Faghri, C.-L. Li, C. Koc, N. True, A. Antony, G. Santhanam, J. Gabriel, P. Grasch, O. Tuzel *et al.*, "Fastvlm: Efficient vision encoding for vision language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 19 769–19 780.

[79] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[80] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International journal of computer vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[81] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[82] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[83] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[84] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[85] P. Belcak, G. Heinrich, S. Diao, Y. Fu, X. Dong, S. Muralidharan, Y. C. Lin, and P. Molchanov, "Small language models are the future of agentic ai," *arXiv preprint arXiv:2506.02153*, 2025.

[86] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018.

[87] D. Rivkin, F. Hogan, A. Feriani, A. Konar, A. Sigal, X. Liu, and G. Dudek, "Aiot smart home via autonomous llm agents," *IEEE Internet of Things Journal*, 2024.

[88] X. Jiang, F. R. Yu, T. Song, and V. C. Leung, "Intelligent resource allocation for video analytics in blockchain-enabled internet of autonomous vehicles with edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 260–14 272, 2020.

[89] Y. Li, J. Sun, Y. Liu, Y. Zhang, A. Li, B. Chen, H. R. Roth, D. Xu, T. Chen, and Y. Chen, "Federated black-box prompt tuning system for large language models on the edge," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 1775–1777.

[90] O. Friha, M. A. Ferrag, B. Kantarci, B. Cakmak, A. Ozgun, and N. Ghoualmi-Zine, "Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness," *IEEE Open Journal of the Communications Society*, 2024.

[91] Y. Li, H. Wen, W. Wang, X. Li, Y. Yuan, G. Liu, J. Liu, W. Xu, X. Wang, Y. Sun *et al.*, "Personal llm agents: Insights and survey about the capability, efficiency and security," *arXiv preprint arXiv:2401.05459*, 2024.

[92] K.-T. Tran, D. Dao, M.-D. Nguyen, Q.-V. Pham, B. O'Sullivan, and H. D. Nguyen, "Multi-agent collaboration mechanisms: A survey of llms," *arXiv preprint arXiv:2501.06322*, 2025.

[93] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, "Efficient acceleration of deep learning inference on resource-constrained edge devices: A review," *Proceedings of the IEEE*, vol. 111, no. 1, pp. 42–91, 2022.

[94] J. Fan, Y. Zhang, X. Li, and D. S. Nikolopoulos, "Parallel cpu-gpu execution for llm inference on constrained gpus," *arXiv preprint arXiv:2506.03296*, 2025.

[95] S. Liu, K. Han, A. Fernandez-Lopez, A. K. Jaiswal, Z. Atashgahi, B. Wu, E. Ponti, C. Hao, R. Burkholz, O. Saukh *et al.*, "Edge-llms: Edge-device large language model competition," in *NeurIPS 2024 Competition Track*, 2024.

[96] X. Li, S. Ghafouri, B. Ji, H. Vandierendonck, D. John, and D. S. Nikolopoulos, "Qpart: Adaptive model quantization and dynamic workload balancing for accuracy-aware edge inference," *arXiv preprint arXiv:2506.23934*, 2025.

[97] A. Ltd., "Arm cortex-a series programmer's guide / performance brief," 2023. [Online]. Available: https://developer.arm.com/documentation/den0013/0400/Preface

[98] F. J. E. N. Andong and Q. Min, "Federated multi-agent reinforcement learning for privacy-preserving and energy-aware resource management in 6g edge networks," *arXiv preprint arXiv:2509.10163*, 2025.

[99] H. Hu, W. Du, Y. Li, and Y. Wang, "pfl-sbpm: A communication-efficient personalized federated learning framework for resource-limited edge clients," *Future Generation Computer Systems*, vol. 171, p. 107849, 2025.

[100] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[101] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799.

[102] K. Le, N. Luong-Ha, M. Nguyen-Duc, D. Le-Phuoc, C. Do, and K.-S. Wong, "Exploring the practicality of federated learning: A survey towards the communication perspective," *arXiv preprint arXiv:2405.20431*, 2024.

[103] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face," *Advances in Neural Information Processing Systems*, vol. 36, pp. 38 154–38 180, 2023.

[104] X. Li, M. Abdallah, S. Suryavansh, M. Chiang, K. T. Kim, and S. Bagchi, "Dag-based task orchestration for edge computing," in *2022 41st International Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 2022, pp. 23–34.

[105] T. Xu, W.-j. Lu, J. Yu, Y. Chen, C. Lin, R. Wang, and M. Li, "Breaking the layer barrier: Remodeling private transformer inference with hybrid {CKKS} and {MPC}," in *34th USENIX Security Symposium (USENIX Security 25)*, 2025, pp. 2653–2672.

[106] Y. Zhang, C. Gu, P. Shi, Z. Jing, B. Li, and B. Liu, "Bring your device group (bydg): Efficient and privacy-preserving user-device authentication protocol in multi-access edge computing," *IEEE Transactions on Information Forensics and Security*, 2025.

[107] Z. Yuan, Y. Shang, Y. Zhou, Z. Dong, Z. Zhou, C. Xue, B. Wu, Z. Li, Q. Gu, Y. J. Lee *et al.*, "Llm inference unveiled: Survey and roofline model insights," *arXiv preprint arXiv:2402.16363*, 2024.

[108] A. Ignatov, R. Timofte, M. Denna, and A. Younes, "Real-time quantized image super-resolution on mobile npus, mobile ai 2021 challenge: Report," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2525–2534.

[109] A. Kouris, S. I. Venieris, S. Laskaridis, and N. D. Lane, "Fluid batching: Exit-aware preemptive serving of early-exit neural networks on edge npus," *arXiv preprint arXiv:2209.13443*, 2022.

[110] 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," in *Proceedings of the 29th symposium on operating systems principles*, 2023, pp. 611–626.

[111] T. Dao, "Flashattention-2: Faster attention with better parallelism and work partitioning," in *12th International Conference on Learning Representations, ICLR 2024*, 2024.

[112] B. Wu, S. Yan, S. Zhang, J. Lu, Y. Zeng, Y. Wang, and X. Zhou, "Efficient pretraining length scaling," *arXiv preprint arXiv:2504.14992*, 2025.

[113] T. H. Sakib, M. T. Hosain, and M. K. Morol, "Small language models: Architectures, techniques, evaluation, problems and future adaptation," *arXiv preprint arXiv:2505.19529*, 2025.

[114] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou, "A survey on knowledge distillation of large language models," *arXiv preprint arXiv:2402.13116*, 2024.

[115] Betterdata, "Data augmentation with synthetic data for ai and ml," 2025, accessed: June 19, 2025. [Online]. Available: https://www.betterdata.ai/blogs/data-augmentation-with-synthetic-data-for-ai-and-ml

[116] AIS, "Knowledge systems and synthetic data: The role of generative ai in data augmentation," 2025, accessed: June 19, 2025. [Online]. Available: https://www.ais.com/knowledge-systems-and-synthetic-data/

[117] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "Mobilebert: A compact task-agnostic bert for resource-limited devices," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 8965–8972.

[118] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[119] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 4163–4174.

[120] K. Tirumala, D. Simig, A. Aghajanyan, and A. Morcos, "D4: Improving llm pretraining via document de-duplication and diversification," *Advances in Neural Information Processing Systems*, vol. 36, pp. 53 983–53 995, 2023.

[121] P. Zhang, G. Zeng, T. Wang, and W. Lu, "Tinyllama: An open-source small language model," *arXiv preprint arXiv:2401.02385*, 2024.

[122] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv e-prints*, pp. arXiv–2407, 2024.

[123] 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.

[124] A. Lozhkov, E. Bakouch, G. M. Blazquez, G. Penedo, L. Tunstall, A. Marafioti, A. P. Lajarín, H. Kydlíček, V. Srivastav, J. Lochner *et al.*, "Smollm2: When smol goes big—data-centric training of a fully open small language model," in *Second Conference on Language Modeling*.

[125] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2704–2713.

[126] S. Laskaridis, K. Katevas, L. Minto, and H. Haddadi, "Mobile and edge evaluation of large language models," in *Workshop on Efficient Systems for Foundation Models II@ ICML2024*, 2024.

[127] J. Xiao, Q. Huang, X. Chen, and C. Tian, "Understanding large language models in your pockets: Performance study on cots mobile devices," *arXiv preprint arXiv:2410.03613*, 2024.

[128] F. Tan, R. Lee, Ł. Dudziak, S. X. Hu, S. Bhattacharya, T. Hospedales, G. Tzimiropoulos, and B. Martinez, "Mobilequant: Mobile-friendly quantization for on-device language models," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 9761–9771.

[129] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han, "Haq: Hardware-aware automated quantization with mixed precision," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8612–8620.

[130] S. Q. Zhang, T. Tambe, N. Cuevas, G.-Y. Wei, and D. Brooks, "Camel: Co-designing ai models and embedded drams for efficient on-device learning," *arXiv preprint arXiv:2305.03148*, 2023.

[131] Z. Lu, X. Li, D. Cai, R. Yi, F. Liu, X. Zhang, N. D. Lane, and M. Xu, "Small language models: Survey, measurements, and insights," *arXiv preprint arXiv:2409.15790*, 2024.

[132] C. Van Nguyen, X. Shen, R. Aponte, Y. Xia, S. Basu, Z. Hu, J. Chen, M. Parmar, S. Kunapuli, J. Barrow *et al.*, "A survey of small language models," *arXiv preprint arXiv:2410.20011*, 2024.

[133] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love *et al.*, "Gemma: Open models based on gemini research and technology," *arXiv preprint arXiv:2403.08295*, 2024.

[134] M. Mishra, M. Stallone, G. Zhang, Y. Shen, A. Prasad, A. M. Soria, M. Merler, P. Selvam, S. Surendran, S. Singh *et al.*, "Granite code models: A family of open foundation models for code intelligence," *arXiv preprint arXiv:2405.04324*, 2024.

[135] T. M. Pham, P. T. Nguyen, S. Yoon, V. D. Lai, F. Dernoncourt, and T. Bui, "Slimlm: An efficient small language model for on-device document assistance," *arXiv preprint arXiv:2411.09944*, 2024.

[136] M. Marone, O. Weller, W. Fleshman, E. Yang, D. Lawrie, and B. Van Durme, "mmbert: A modern multilingual encoder with annealed language learning," *arXiv preprint arXiv:2509.06888*, 2025.

[137] C. Zhao, E. Chang, Z. Liu, C.-J. Chang, W. Wen, C. Lai, R. Cao, Y. Tian, R. Krishnamoorthi, Y. Shi *et al.*, "Mobilellm-r1: Exploring the limits of sub-billion language model reasoners with open training recipes," *arXiv preprint arXiv:2509.24945*, 2025.

[138] R. Yi, L. Guo, S. Wei, A. Zhou, S. Wang, and M. Xu, "Edgemoe: Fast on-device inference of moe-based large language models," *arXiv preprint arXiv:2308.14352*, 2023.

[139] ——, "Edgemoe: Empowering sparse large language models on mobile devices," *IEEE Transactions on Mobile Computing*, 2025.

[140] J. Li, Z. Sun, X. He, L. Zeng, Y. Lin, E. Li, B. Zheng, R. Zhao, and X. Chen, "Locmoe: a low-overhead moe for large language model training," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024, pp. 6377–6387.

[141] Y. Shen, Z. Guo, T. Cai, and Z. Qin, "Jetmoe: Reaching llama2 performance with 0.1 m dollars," *arXiv preprint arXiv:2404.07413*, 2024.

[142] X. Yao, H. Qian, X. Hu, G. Xu, W. Liu, J. Luan, B. Wang, and Y. Liu, "Theoretical insights into fine-tuning attention mechanism: Generalization and optimization," *arXiv preprint arXiv:2410.02247*, 2024.

[143] H. Cai, C. Gan, L. Zhu, and S. Han, "Tinytl: Reduce memory, not parameters for efficient on-device learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 285–11 297, 2020.

[144] S. Cai, X. Liu, J. Yuan, and Q. Zhou, "Prompt-ladder: Memory-efficient prompt tuning for vision-language models on edge devices," *Pattern Recognition*, vol. 163, p. 111460, 2025.

[145] Aisera, "What are small language models (slms)?" 2025, accessed: June 19, 2025. [Online]. Available: https://aisera.com/blog/small-language-models/

[146] M. AI, "Llama 3.1 8b," 2024, available at: https://llama.meta.com/.

[147] F. Meng, C.-A. Wang, and L. Zhang, "Evolution and efficiency in neural architecture search: Bridging the gap between expert design and automated optimization," *arXiv preprint arXiv:2403.17012*, 2024.

[148] X. Wang, H. Shangguan, F. Huang, S. Wu, and W. Jia, "Mel: Efficient multi-task evolutionary learning for high-dimensional feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 8, pp. 4020–4033, 2024.

[149] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2820–2828.

[150] S. Gupta, "Neural architecture search for ai model optimization," *International Journal of Artificial Intelligence and Machine Learning*, vol. 4, no. 2, 2017.

[151] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *International Conference on Learning Representations*, 2019.

[152] B. Wu, X. Dai, P. Zhang, Y. Wang, F. Sun, Y. Wu, Y. Tian, P. Vajda, Y. Jia, and K. Keutzer, "Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 734–10 742.

[153] X. Wang, Y. Wang, K.-C. Wong, and X. Li, "A self-adaptive weighted differential evolution approach for large-scale feature selection," *Knowledge-Based Systems*, vol. 235, p. 107633, 2022.

[154] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *ICLR*, 2016.

[155] X. Ma, G. Fang, and X. Wang, "Llm-pruner: On the structural pruning of large language models," *Advances in neural information processing systems*, vol. 36, pp. 21 702–21 720, 2023.

[156] E. Frantar and D. Alistarh, "Sparsegpt: Massive language models can be accurately pruned in one-shot," in *International conference on machine learning*. PMLR, 2023, pp. 10 323–10 337.

[157] NVIDIA, "Llm model pruning and knowledge distillation with nvidia nemo framework," https://developer.nvidia.com/blog/llm-model-pruning-and-knowledge-distillation-with-nvidia-nemo-framework/, 2025, accessed: June 19, 2025.

[158] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2020.

[159] W. Zhao, T. Pan, X. Han, Y. Zhang, A. Sun, Y. Huang, K. Zhang, W. Zhao, Y. Li, J. Wang *et al.*, "Fr-spec: Accelerating large-vocabulary language models via frequency-ranked speculative sampling," *arXiv preprint arXiv:2502.14856*, 2025.

[160] Y. Li, F. Wei, C. Zhang, and H. Zhang, "Eagle-2: Faster inference of language models with dynamic draft trees," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 7421–7432.

[161] F. Gloeckle, B. Y. Idrissi, B. Roziere, D. Lopez-Paz, and G. Synnaeve, "Better & faster large language models via multi-token prediction," in *International Conference on Machine Learning*. PMLR, 2024, pp. 15 706–15 734.

[162] Y. Zhang, W. Zhao, X. Han, T. Zhao, W. Xu, H. Cao, and C. Zhu, "Speculative decoding meets quantization: Compatibility evaluation and hierarchical framework design," *arXiv preprint arXiv:2505.22179*, 2025.

[163] Symbl.ai, "A guide to quantization in llms," 2025, accessed: June 19, 2025. [Online]. Available: https://symbl.ai/developers/blog/a-guide-to-quantization-in-llms/

[164] H. Chen, Y. Wen, L. Cheng, S. Kuang, Y. Liu, W. Li, L. Li, R. Zhang, X. Song, W. Li *et al.*, "Autoos: make your os more powerful by exploiting large language models," in *Forty-first International Conference on Machine Learning*, 2024.

[165] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in neural information processing systems*, vol. 36, pp. 10 088–10 115, 2023.

[166] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "Gptq: Accurate post-training quantization for generative pre-trained transformers," in *The Eleventh International Conference on Learning Representations*. OpenReview, 2023.

[167] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, "Awq: Activation-aware weight quantization for on-device llm compression and acceleration," *Proceedings of machine learning and systems*, vol. 6, pp. 87–100, 2024.

[168] Y. Li, R. Gong, X. Tan, Y. Yang, P. Hu, Q. Zhang, F. Yu, W. Wang, and S. Gu, "Brecq: Pushing the limit of post-training quantization by block reconstruction," in *International Conference on Learning Representations*, 2021.

[169] W. Chen, Z. Li, and S. Xin, "Omnivlm: A token-compressed, sub-billion-parameter vision-language model for efficient on-device inference," *arXiv preprint arXiv:2412.11475*, 2024.

[170] Z. Ravichandran, I. Hounie, F. Cladera, A. Ribeiro, G. J. Pappas, and V. Kumar, "Distilling on-device language models for robot planning with minimal human intervention," *arXiv preprint arXiv:2506.17486*, 2025.

[171] O. Bohdal, M. Ozay, J. Moon, K.-H. Lee, H. Ko, and U. Michieli, "Efficient compositional multi-tasking for on-device large language models," *arXiv preprint arXiv:2507.16083*, 2025.

[172] M. Huang, R. Huang, H. Shi, Y. Chen, C. Zheng, X. Sun, X. Jiang, Z. Li, and H. Cheng, "Efficient multi-modal large language models via visual token grouping," *arXiv preprint arXiv:2411.17773*, 2024.

[173] K. Cai, Z. Duan, G. Liu, C. Fleming, and C. X. Lu, "Self-adapting large visual-language models to edge devices across visual modalities," in *European Conference on Computer Vision*. Springer, 2024, pp. 301–318.

[174] M. Yang, Z. Jia, Z. Dai, S. Guo, and L. Wang, "Mobileviclip: An efficient video-text model for mobile devices," *arXiv preprint arXiv:2508.07312*, 2025.

[175] 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," 2024.

[176] M. Team, C. Xiao, Y. Li, X. Han, Y. Bai, J. Cai, H. Chen, W. Chen, X. Cong, G. Cui *et al.*, "Minicpm4: Ultra-efficient llms on end devices," *arXiv preprint arXiv:2506.07900*, 2025.

[177] Walturn, "What is ollama? features, pricing, and use cases," 2025, accessed: June 19, 2025. [Online]. Available: https://www.walturn.com/insights/what-is-ollama-features-pricing-and-use-cases

[178] M. LLM, "Mlc llm | home," 2025, accessed: June 19, 2025. [Online]. Available: https://llm.mlc.ai/

[179] X. Lu, Y. Chen, C. Chen, H. Tan, B. Chen, Y. Xie, R. Hu, G. Tan, R. Wu, Y. Hu *et al.*, "Bluelm-v-3b: Algorithm and system co-design for multimodal large language models on mobile devices," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 4145–4155.

[180] H. Cai, J. Lin, Y. Lin, Z. Liu, H. Tang, H. Wang, L. Zhu, and S. Han, "Enable deep learning on mobile devices: Methods, systems, and applications," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 27, no. 3, pp. 1–50, 2022.

[181] Y. Huang, J. Xu, B. Pei, L. Yang, M. Zhang, Y. He, G. Chen, X. Chen, Y. Wang, Z. Nie *et al.*, "Vinci: A real-time smart assistant based on egocentric vision-language model for portable devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 9, no. 3, pp. 1–33, 2025.

[182] A. Abouelenin, A. Ashfaq, A. Atkinson, H. Awadalla, N. Bach, J. Bao, A. Benhaim, M. Cai, V. Chaudhary, C. Chen *et al.*, "Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras," *arXiv preprint arXiv:2503.01743*, 2025.

[183] Y. Peng, G. Zhang, M. Zhang, Z. You, J. Liu, Q. Zhu, K. Yang, X. Xu, X. Geng, and X. Yang, "Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl," *arXiv preprint arXiv:2503.07536*, 2025.

[184] F. Faghri, P. K. A. Vasu, C. Koc, V. Shankar, A. Toshev, O. Tuzel, and H. Pouransari, "Mobileclip2: Improving multi-modal reinforced training," *arXiv preprint arXiv:2508.20691*, 2025.

[185] L. Rockchip Electronics Co., "Rk3588 - rockchip flagship soc for aiot applications," https://www.rock-chips.com/a/cn/product/RK35xilie/2022/0926/1656.html, 2024, 8nm process flagship chip with 6 TOPS NPU and 8K video capabilities.

[186] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, vol. 1, no. 2, 2023.

[187] L. Rockchip Electronics Co., "Rockchip 2024 annual report: Aiot soc market leadership and automotive breakthrough," Rockchip Electronics, Tech. Rep., 2024, comprehensive analysis of financial performance and strategic positioning in AIoT and automotive markets.

[188] L. Zheng, L. Yin, Z. Xie, C. L. Sun, J. Huang, C. H. Yu, S. Cao, C. Kozyrakis, I. Stoica, J. E. Gonzalez *et al.*, "Sglang: Efficient execution of structured language model programs," *Advances in neural information processing systems*, vol. 37, pp. 62 557–62 583, 2024.

[189] G. Yang, C. Demirkiran, Z. E. Kizilates, C. A. R. Ocampo, A. K. Coskun, and A. Joshi, "Processing-in-memory using optically-addressed phase change memory," in *2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2023, pp. 1–6.

[190] Z. Zhou, J. Liu, Z. Gu, and G. Sun, "Energon: Toward efficient acceleration of transformers using dynamic sparse attention," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 1, pp. 136–149, 2022.

[191] D. Kudithipudi, C. Schuman, C. M. Vineyard, T. Pandit, C. Merkel, R. Kubendran, J. B. Aimone, G. Orchard, C. Mayr, R. Benosman *et al.*, "Neuromorphic computing at scale," *Nature*, vol. 637, no. 8047, pp. 801–812, 2025.

[192] I. Schuler and R. Stevens, "Neuromorphic computing: From materials to systems architecture," *Office of Scientific and Technical Information*, 2015.

[193] S. Alabed, D. Grewe, N. A. Rink, T. Sitdikov, A. Swietlik, D. Vytiniotis, and D. Belov, "Toast: Fast and scalable auto-partitioning based on principled static analysis," *arXiv preprint arXiv:2508.15010*, 2025.

[194] L. Rockchip Electronics Co., "Rk182x series: Edge ai co-processors for local llm deployment," https://www.rock-chips.com, 2024, dedicated AI co-processors enabling efficient on-device large language model inference.

[195] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis, "Efficient streaming language models with attention sinks," in *The Twelfth International Conference on Learning Representations*, 2024.

[196] Z. Cai, R. Ma, Y. Fu, W. Zhang, R. Ma, and H. Guan, "Llmaas: Serving large language models on trusted serverless computing platforms," *IEEE Transactions on Artificial Intelligence*, 2024.

[197] Y. Wu, R. Chen, P. Liu, and H. Qian, "Livelongbench: Tackling long-context understanding for spoken texts from live streams," *arXiv preprint arXiv:2504.17366*, 2025.

[198] Y. Sheng, L. Zheng, B. Yuan, Z. Li, M. Ryabinin, B. Chen, P. Liang, C. Ré, I. Stoica, and C. Zhang, "Flexgen: High-throughput generative inference of large language models with a single gpu," in *International Conference on Machine Learning*. PMLR, 2023, pp. 31 094–31 116.

[199] J. Zhao, W. Lu, S. Wang, L. Kong, and C. Wu, "Qspec: Speculative decoding with complementary quantization schemes," *arXiv preprint arXiv:2410.11305*, 2024.

[200] Z. Xue, Y. Song, Z. Mi, X. Zheng, Y. Xia, and H. Chen, "Powerinfer-2: Fast large language model inference on a smartphone," *arXiv preprint arXiv:2406.06282*, 2024.

[201] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "The sparsely-gated mixture-of-experts layer," *Outrageously large neural networks*, vol. 2, 2017.

[202] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[203] M. Ali, F. Naeem, M. Tariq, and G. Kaddoum, "Federated learning for privacy preservation in smart healthcare systems: A comprehensive survey," *IEEE journal of biomedical and health informatics*, vol. 27, no. 2, pp. 778–789, 2022.

[204] K. Masayoshi, M. Hashimoto, R. Yokoyama, N. Toda, Y. Uwamino, S. Fukuda, H. Namkoong, and M. Jinzaki, "Ehr-mcp: Real-world evaluation of clinical information retrieval by large language models via model context protocol," 2025. [Online]. Available: https://arxiv.org/abs/2509.15957

[205] C. Yu, Y. Zhang, Z. Liu, Z. Ding, Y. Sun, and Z. Jin, "Frame: Feedback-refined agent methodology for enhancing medical research insights," *arXiv preprint arXiv:2505.04649*, 2025.

[206] C. Liu, D. Li, Y. Shu, R. Chen, D. Duan, T. Fang, and B. Dai, "Fleming-r1: Toward expert-level medical reasoning via reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2509.15279

[207] T. Cai, Y. Liu, Z. Zhou, H. Ma, S. Z. Zhao, Z. Wu, and J. Ma, "Driving with regulation: Interpretable decision-making for autonomous

vehicles with retrieval-augmented reasoning via llm," *arXiv preprint arXiv:2410.04759*, 2024.
[208] Y. Huang, J. Sansom, Z. Ma, F. Gervits, and J. Chai, "Drivlme: Enhancing llm-based autonomous driving agents with embodied and social experiences," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 3153–3160.
[209] Y. Hu, F. Wang, D. Ye, M. Wu, J. Kang, and R. Yu, "Llm-based misbehavior detection architecture for enhanced traffic safety in connected autonomous vehicles," *IEEE Transactions on Vehicular Technology*, 2025.
[210] H.-k. Chiu, R. Hachiuma, C.-Y. Wang, S. F. Smith, Y.-C. F. Wang, and M.-H. Chen, "V2v-llm: Vehicle-to-vehicle cooperative autonomous driving with multi-modal large language models," *arXiv preprint arXiv:2502.09980*, 2025.
[211] A. Gueriani, H. Kheddar, A. C. Mazari, and M. C. Ghanem, "A robust cross-domain ids using bigru-lstm-attention for medical and industrial iot security," *ICT Express*, 2025.
[212] H. Yang, Y. Zhou, T. Liang, and L. Kuang, "Chatdl: An llm-based defect localization approach for software in iiot flexible manufacturing," *IEEE Internet of Things Journal*, 2025.
[213] J. Tang, J. Chen, J. He, F. Chen, Z. Lv, G. Han, Z. Liu, H. H. Yang, and W. Li, "Towards general industrial intelligence: A survey of large models as a service in industrial iot," *IEEE Communications Surveys & Tutorials*, 2025.
[214] Z. Li, L. Xia, X. Ren, J. Tang, T. Chen, Y. Xu, and C. Huang, "Urban computing in the era of large language models," *ACM Transactions on Intelligent Systems and Technology*, 2025.
[215] Y. Zheng, F. Xu, Y. Lin, P. Santi, C. Ratti, Q. R. Wang, and Y. Li, "Urban planning in the era of large language models," *Nature Computational Science*, pp. 1–10, 2025.
[216] B. Yang, Y. Zhang, L. Feng, Y. Chen, J. Zhang, X. Xu, N. Aierken, Y. Li, Y. Chen, G. Yang *et al.*, "Agrigpt: a large language model ecosystem for agriculture," *arXiv preprint arXiv:2508.08632*, 2025.
[217] Z. Yuan, K. Liu, S. Li, R. Peng, D. Leybourne, N. Musa, and P. Yang, "Pezego: A precision agriculture system based on large language models and internet of things for pest management," *IEEE Internet of Things Journal*, 2025.
[218] Y. Zhou and M. Ryo, "Agribench: A hierarchical agriculture benchmark for multimodal large language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 207–223.
[219] A. Tzachor, M. Devare, C. Richards, P. Pypers, A. Ghosh, J. Koo, S. Johal, and B. King, "Large language models and agricultural extension services," *Nature food*, vol. 4, no. 11, pp. 941–948, 2023.
[220] Z. Li, B. Wu, Y. Zhang, X. Li, K. Li, and W. Chen, "Cusmer: Multimodal intent recognition in customer service via data augment and llm merge," in *Companion Proceedings of the ACM on Web Conference 2025*, 2025, pp. 3058–3062.
[221] S. Farfade, S. Vernekar, V. Chaoji, and R. Mukherjee, "Scaling use-case based shopping using llms," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 2024, pp. 1165–1166.
[222] H. Zhang, X. Kang, and J. Guo, "How does search affect personalized recommendations and user behavior? evidence from llm-based synthetic data," in *Companion Proceedings of the ACM on Web Conference 2025*, 2025, pp. 2434–2443.
[223] Y. Zhao, X. Hou, S. Wang, and H. Wang, "Llm app store analysis: A vision and roadmap," *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 5, pp. 1–25, 2025.
[224] S. Mao, L. Cheng, P. Cai, G. Yan, D. Wang, and B. Shi, "Deepwriter: A fact-grounded multimodal writing assistant based on offline knowledge base," *arXiv preprint arXiv:2507.14189*, 2025.
[225] 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.
[226] C. F. Ruan, Y. Qin, X. Zhou, R. Lai, H. Jin, Y. Dong, B. Hou, M.-S. Yu, Y. Zhai, S. Agarwal *et al.*, "Webllm: A high-performance in-browser llm inference engine," *arXiv preprint arXiv:2412.15803*, 2024.
[227] L. Rockchip Electronics Co., "Rk3588 brief datasheet," https://www.rock-chips.com/uploads/pdf/2022.8.26/192/RK3588BriefDatasheet.pdf, Rockchip Electronics, Datasheet, 2022.
[228] C. Software, "Rockchip rk3588 soc datasheet reveals 6 tops npu and 8k video capabilities," *CNX Software*, 2021, online article.
[229] I. I. Corp., "Wafer-rk3588 industrial sbc specification (rk3588, 6 tops npu, mixed precision)," https://www.ieiworld.com/en/product/model.php?II=1036, 2024, industrial SBC Spec Sheet.
[230] G. G. et al., "`llama.cpp`: Local inference of llms on diverse hardware," https://github.com/ggml-org/llama.cpp, 2023.
[231] H. Chen, C. Tian, Z. He, B. Yu, Y. Liu, and J. Cao, "Inference performance evaluation for llms on edge devices with a novel benchmarking framework and metric," *arXiv preprint arXiv:2508.11269*, 2025.
[232] Y. Cheng, M. Xu, Y. Zhang, K. Li, R. Wang, and L. Yang, "Autoiot: Automated iot platform using large language models," *IEEE Internet of Things Journal*, 2024.
[233] A. Joshi, S. Sanyal, and K. Roy, "Neuro-lift: A neuromorphic, llm-based interactive framework for autonomous drone flight at the edge," *arXiv preprint arXiv:2501.19259*, 2025.
[234] R. Li, W. Wei, Q. Xin, X. Liu, S. Mao, E. Ma, Z. Chen, M. Zhang, H. Li, and Z. Zhang, "What is next for llms? next-generation ai computing hardware using photonic chips," *arXiv preprint arXiv:2505.05794*, 2025.
[235] X. Kong, L. Li, Z. Chen, C. Xue, X. Xu, H. Liu, Y. Wu, Y. Fang, H. Fang, K. Chen *et al.*, "Quantum-enhanced llm efficient fine tuning," *arXiv preprint arXiv:2503.12790*, 2025.
[236] J. Jin, Y. Zhang, R. Xu, and Y. Chen, "An innovative brain-computer interface interaction system based on the large language model," *arXiv preprint arXiv:2502.11659*, 2025.
[237] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and trends® in theoretical computer science*, vol. 9, no. 3–4, pp. 211–407, 2014.
[238] Y. Boo, S. Shin, J. Choi, and W. Sung, "Stochastic precision ensemble: self-knowledge distillation for quantized deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6794–6802.
[239] L. Zhao, Y. Zhang, and J. Yang, "Sca: a secure cnn accelerator for both training and inference," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.
[240] Y. Cui, S. Wu, Q. Li, A. B. Chan, T.-W. Kuo, and C. J. Xue, "Bits-ensemble: Toward light-weight robust deep ensemble by bits-sharing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4397–4408, 2022.
[241] M. Alsuleman, P. Duncan, and A. Thompson, "Screening of atrial fibrillation using wearable ppg devices–a trustworthy and safe ai life cycle case study," 2025.
[242] TTMS, "Iso 27001 implementation - improve data security," 2025, accessed: June 19, 2025. [Online]. Available: https://ttms.com/iso-27001-implementation-strengthen-data-security-in-your-company/
[243] HPE, "What is ai security | glossary," 2025, accessed: June 19, 2025. [Online]. Available: https://www.hpe.com/us/en/what-is/ai-security.html
[244] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for ai: Review and open research challenges," *IEEE access*, vol. 7, pp. 10 127–10 149, 2019.
[245] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 188–19 209, 2024.
[246] Y. Liu, B. Yan, T. Zou, J. Zhang, Z. Gu, J. Ding, X. Wang, J. Li, X. Ye, Y. Ouyang *et al.*, "Towards harnessing the collaborative power of large and small models for domain tasks," *arXiv preprint arXiv:2504.17421*, 2025.
[247] W. Chen, Z. Zhao, J. Yao, Y. Zhang, J. Bu, and H. Wang, "Multi-modal medical diagnosis via large-small model collaboration," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 30 763–30 773.
[248] Z. Liu, K. Liu, M. Guo, S. Zhang, and Y. Wang, "Cotuning: A large-small model collaborating distillation framework for better model generalization," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 10 487–10 496.
[249] H. Wang, L. Ren, T. Zhao, and L. Jiao, "Collm: Industrial large-small model collaboration with fuzzy decision-making agent and self-reflection," *IEEE Transactions on Fuzzy Systems*, 2025.
[250] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Edge intelligence for energy-efficient computation offloading and resource allocation in 5g beyond," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12 175–12 186, 2020.
[251] Y. Tian, Z. Zhang, Y. Yang, Z. Chen, Z. Yang, R. Jin, T. Q. Quek, and K.-K. Wong, "An edge-cloud collaboration framework for generative ai service provision with synergetic big cloud model and small edge models," *IEEE Network*, vol. 38, no. 5, pp. 37–46, 2024.
[252] R. Murthy, L. Yang, J. Tan, T. M. Awalgaonkar, Y. Zhou, S. Heinecke, S. Desai, J. Wu, R. Xu, S. Tan *et al.*, "Mobileaibench: Benchmarking llms and lmms for on-device use cases," *arXiv preprint arXiv:2406.10290*, 2024.
[253] E. Frantar, R. L. Castro, J. Chen, T. Hoefler, and D. Alistarh, "Marlin: Mixed-precision auto-regressive parallel inference on large language models," in *Proceedings of the 30th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, 2025, pp. 239–251.

[254] ISO/IEC, "Iso/iec 9126-1:2001 software engineering – product quality – part 1: Quality model," 2001.
[255] J. Wang, M. Wang, Y. Zhou, Z. Xing, Q. Liu, X. Xu, W. Zhang, and L. Zhu, "Llm-based hse compliance assessment: Benchmark, performance, and advancements," *arXiv preprint arXiv:2505.22959*, 2025.
[256] Y. Fang, "Deep learning on the edge: Model partitioning, caching, and compression," McMaster University Institutional Repository (MacSphere), 2025, accessed: June 19, 2025. [Online]. Available: https://macsphere.mcmaster.ca/handle/11375/25576
[257] G. Zhang, W. Guo, Z. Tan, and H. Jiang, "Amp4ec: Adaptive model partitioning framework for efficient deep learning inference in edge computing environments," *arXiv preprint arXiv:2504.00407*, 2025.
[258] Atos, "Neuromorphic computing: The future of ai and beyond," 2025, accessed: June 19, 2025. [Online]. Available: https://atos.net/en/blog/neuromorphic-computing-the-future-of-ai-and-beyond
[259] C. Xiao, P. Zhang, X. Han, G. Xiao, Y. Lin, Z. Zhang, Z. Liu, and M. Sun, "Infllm: Training-free long-context extrapolation for llms with an efficient context memory," *Advances in Neural Information Processing Systems*, vol. 37, pp. 119 638–119 661, 2024.

Xubin Wang (Student Member, IEEE) is currently with the Beijing Normal-Hong Kong Baptist University and the Institute of Artificial Intelligence and Future Networks, Beijing Normal University (Zhuhai campus). His research combines evolutionary optimization, multi-task feature selection, and reinforcement learning with representation compression to enhance generalization across heterogeneous, high-dimensional biomedical and textual datasets. Representative work spans large model prompting and selection, evolutionary feature learning, biomedical classification and pathway modeling, and edge/on-device inference. He has published in ICML, IEEE TKDE, IEEE TCBB, ACM CSUR, and Knowledge-Based Systems, with several results receiving media coverage. His current research focuses on synergistic LLM–edge co-training, pathway-level mechanistic modeling, and reliable evaluation protocols for collaborative small–large model systems. He is a student member of IEEE.

Qing Li (Fellow, IEEE) received the BEng degree from Hunan University, Changsha, China, and the MSc and PhD degrees from the University of Southern California, Los Angeles, all in computer science. He is currently a Chair Professor (Data Science) and the Head of the Department of Computing at The Hong Kong Polytechnic University. He is a Fellow of IEEE and a member of ACM SIGMOD and IEEE Technical Committee on Data Engineering. His research interests include object modeling, multimedia databases, social media, and recommender systems. He has been actively involved in the research community by serving as an associate editor and reviewer for technical journals, and as an organizer/co-organizer of numerous international conferences. He is the chairperson of the Hong Kong Web Society, and also served/is serving as an executive committee (EXCO) member of IEEE-Hong Kong Computer Chapter and ACM Hong Kong Chapter. In addition, he serves as a councilor of the Database Society of Chinese Computer Federation (CCF), a member of the Big Data Expert Committee of CCF, and is a Steering Committee member of DASFAA, ER, ICWL, UMEDIA, and WISE Society.

Weijia Jia (Fellow, IEEE) is currently a Chair Professor, Director of the BNU-BNBU Institute of Artificial Intelligence and Future Networks at Beijing Normal University (Zhuhai), and used to be VP for Research of Beijing Normal-Hong Kong Baptist University. He has also held the Zhiyuan Chair Professor position at Shanghai Jiao Tong University, China. Previously, he was Chair Professor and Deputy Director of the State Key Laboratory of Internet of Things for Smart City at the University of Macau. He earned his BSc (1982) and MSc (1984) from Central South University, China, and his Master of Applied Science/PhD (1992/1993) from the Polytechnic Faculty of Mons, Belgium, all in computer science. From 1993 to 1995 he worked as a research fellow at the German National Research Center for Information Science (GMD) in Bonn (St. Augustine). From 1995 to 2013 he served as a professor at City University of Hong Kong. His contributions span optimal network routing and deployment, anycast and QoS routing, sensor networking, AI (knowledge-relation extraction, NLP, etc.), and edge computing. He has authored over 700 publications in prestigious international journals, conferences, research books, and book chapters. Awards include the Best Product Awards at the International Science and Tech Expo (Shenzhen) in 2011 and 2012, and the 1st Prize of Scientific Research Awards from the Ministry of Education of China in 2017. He has served as area editor for several leading international journals, and as chair, program-committee member, and keynote speaker for many top conferences. He is a Fellow of IEEE and a Distinguished Member of the China Computer Federation (CCF).