Journal of Edge Computing, 2025, Vol. **4**, Iss. 2, pp. 179–233 https://doi.org/10.55056/jec.1000

Edge intelligence unleashed: a survey on deploying large language models in resource-constrained environments

Serhiy O. Semerikov[1,2,3,4,5], Tetiana A. Vakaliuk[4,3,1,5], Olga B. Kanevska[1], Oksana A. Ostroushko[1] and Andrii O. Kolhatin[1]

[1] *Kryvyi Rih State Pedagogical University, 54 Universytetskyi Ave., Kryvyi Rih, 50086, Ukraine*
[2] *Kryvyi Rih National University, 11 Vitalii Matusevych Str., Kryvyi Rih, 50027, Ukraine*
[3] *Institute for Digitalisation of Education of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine*
[4] *Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine*
[5] *Academy of Cognitive and Natural Sciences, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine*

Abstract. Edge computing environments face unprecedented challenges in deploying large language models due to severe resource constraints, latency requirements, and privacy concerns that traditional cloud-based solutions cannot address. Current approaches struggle with the fundamental mismatch between LLMs' computational demands – requiring gigabytes of memory and billions of operations – and edge devices' limited capabilities, resulting in either degraded performance or infeasible deployments. This survey presents a systematic analysis of emerging techniques that enable efficient LLM deployment at the edge through four complementary strategies: model compression via quantisation and pruning that reduces memory footprint by up to 75% while maintaining accuracy, knowledge distillation frameworks achieving 4000× parameter reduction with comparable performance, edge-cloud collaborative architectures like EdgeShard delivering 50% latency reduction through intelligent workload distribution, and hardware-specific optimisations leveraging specialised accelerators. Extensive evaluation across multiple real-world testbeds demonstrates that hybrid edge-microservices architectures achieve 46% lower P99 latency and 67% higher throughput compared to monolithic approaches, while supporting 10,000 concurrent users with 100 ms latency constraints and reducing bandwidth consumption by 99.5% through selective cloud offloading. These advancements enable transformative applications in healthcare monitoring, autonomous systems, real-time IoT analytics, and personalised AI services, fundamentally reshaping how intelligence is delivered at the network edge while preserving privacy and ensuring responsiveness critical for next-generation computing paradigms.[1]

Keywords: edge computing, edge intelligence, large language models, model quantisation, knowledge distillation, distributed inference, real-time inference

1. Introduction

The proliferation of edge computing infrastructure, projected to reach 75 billion connected devices by 2025 [18, 110], has created an urgent need for deploying so-

[1] This is revised and expanded version of the paper [114] presented at the 5th Edge Computing Workshop (doors 2025).

0000-0003-0789-0272 (S. O. Semerikov); 0000-0001-6825-4697 (T. A. Vakaliuk); 0000-0003-1703-7929 (O. B. Kanevska); 0000-0002-5329-8387 (O. A. Ostroushko); 0000-0002-3125-3137 (A. O. Kolhatin)
semerikov@gmail.com (S. O. Semerikov); tetianavakaliuk@acnsci.org (T. A. Vakaliuk); o.b.kanevska@gmail.com (O. B. Kanevska); ostroushko.oksana@gmail.com (O. A. Ostroushko); kolhatin.a@gmail.com (A. O. Kolhatin)
https://kdpu.edu.ua/semerikov (S. O. Semerikov); http://acnsci.org/vakaliuk/ (T. A. Vakaliuk); https://kdpu.edu.ua/personal/obkanevska.html (O. B. Kanevska); https://kdpu.edu.ua/personal/oaostroushko.html (O. A. Ostroushko)

phisticated artificial intelligence capabilities directly at the network periphery. Recent advances demonstrate that 85% of enterprise data will be generated and processed outside traditional data centres by 2026 [58, 171], yet current large language models require computational resources exceeding typical edge device capabilities by three orders of magnitude. Edge environments demand sub-100 ms response times for real-time applications while operating within power budgets below 10 watts, constraints that fundamentally challenge conventional LLM architectures designed for cloud deployment [9, 136].

The deployment of large language models on edge devices presents a multi-dimensional optimisation challenge involving memory constraints, computational limitations, and stringent latency requirements that existing solutions inadequately address. Modern LLMs, even relatively compact variants with 3-7 billion parameters, require 12-28 GB of memory for inference, far exceeding the 2-8 GB available on typical edge devices [56, 139]. The computational complexity of transformer architectures, requiring $O(n^2)$ operations for sequence length n, creates prohibitive latency when executed on edge processors lacking specialised acceleration [67]. Recent attempts by Khalfi and Tabbiche [64] and Bin Son et al. [10] to address these challenges through naive model compression result in accuracy degradation exceeding 30%, while edge-only processing approaches fail to meet real-time constraints for sequences beyond 512 tokens.

Current methodologies for edge LLM deployment fall into three categories, each with critical shortcomings that prevent practical adoption. Cloud-dependent approaches [135, 165] introduce latency penalties of 200-500 ms and privacy vulnerabilities through continuous data transmission, violating edge applications' performance and security requirements. Static compression techniques, including uniform quantisation and structured pruning [90, 139], achieve memory reduction but suffer from catastrophic accuracy loss on domain-specific tasks, particularly affecting multilingual capabilities by up to 25.5 percentage points. Hybrid strategies attempting to balance edge and cloud processing [87, 171] lack dynamic adaptation mechanisms, resulting in suboptimal resource utilisation under varying workload conditions and network states, with throughput degradation of 40% during peak loads compared to theoretically optimal scheduling.

This survey introduces a comprehensive framework for edge LLM deployment that synergistically combines adaptive model optimisation, intelligent workload distribution, and hardware-aware execution strategies to overcome fundamental resource-performance trade-offs. Our analysis reveals that coordinated application of quantisation-aware training, dynamic knowledge distillation, and multi-tier edge-cloud collaboration enables the deployment of 3B-parameter models on devices with 4 GB memory while maintaining 95% of baseline accuracy. The framework's distinguishing innovation lies in its context-aware adaptation mechanism that dynamically adjusts compression levels, partitioning boundaries, and execution strategies based on real-time resource availability and application requirements.

This work makes four principal contributions to edge intelligence research: (1) comprehensive taxonomy of edge LLM optimisation techniques with quantitative performance characterisation across 23 quantisation levels and six model architectures, (2) novel evaluation methodology incorporating latency, throughput, memory efficiency, and energy consumption metrics validated on real-world edge testbeds, (3) systematic analysis of edge-cloud collaborative frameworks including EdgeShard, PAC, and hybrid microservices architectures demonstrating 46-67% performance improvements, and (4) identification of critical research gaps in multi-modal processing, federated learning integration, and hardware-software co-design that define the trajectory of edge LLM development. These contributions establish the theoretical foundation and practical guidelines for deploying intelligence at the network edge.

The remainder of this paper is organised as follows. Section 2 presents background on LLM architectures and edge computing constraints, establishing the fundamental challenges. Section 3 analyses edge-cloud collaborative architectures and dynamic resource allocation strategies. Section 4 explores hardware accelerators and specialised edge processors. Section 5 demonstrates real-world applications across healthcare, IoT, and autonomous systems. Section 6 discusses open challenges and future research directions. Section 7 concludes with implications for edge intelligence deployment.

2. Overview of LLMs for edge deployment

The deployment of large language models on resource-constrained edge devices represents a fundamental shift in distributed artificial intelligence architectures, driven by stringent latency requirements (sub-100 ms for real-time applications), privacy regulations, and bandwidth limitations [12, 23, 107]. Recent empirical studies demonstrate that edge deployment reduces inference latency by 73% compared to cloud-based alternatives while consuming 45% less network bandwidth [105, 109]. The heterogeneous landscape of edge computing platforms – ranging from mobile devices with 4-8 GB RAM to industrial edge servers with specialised accelerators – necessitates sophisticated optimisation strategies that balance computational efficiency with model performance [9, 33].

Contemporary edge deployment paradigms encompass three principal architectures: standalone edge inference, collaborative edge-cloud systems, and peer-to-peer distributed frameworks. Standalone approaches, exemplified by recent implementations on RISC-V platforms [79], achieve inference speeds of 21.77 tokens/second for 2-billion parameter models through vector extension optimisation. Collaborative systems partition computational workloads between edge and cloud resources, with Zhang et al. [171] demonstrating 50% latency reduction through intelligent model sharding. Peer-to-peer frameworks, notably P2PLLMEdge [109], eliminate cloud dependency, achieving a 44.7% reduction in processing duration through decentralised task distribution across heterogeneous devices.

The deployment of LLMs on edge platforms confronts four fundamental technical barriers that significantly impact system design and optimisation strategies. Model size represents the primary constraint, as contemporary LLMs require memory footprints ranging from 350 GB for GPT-3's 175 billion parameters to 4 GB for compressed 7-billion parameter variants [12, 171]. Computational complexity manifests through the quadratic scaling of attention mechanisms with sequence length, requiring $O(n^2d)$ operations where n denotes sequence length and d represents model dimensionality [75, 131]. The self-attention computation alone consumes 67% of inference time on mobile processors, as measured across diverse hardware platforms including Snapdragon 8 Gen 2 and Apple M1 chips [117].

Energy consumption and thermal management are critical factors in sustained edge deployment scenarios. Du et al. [26] reports that continuous LLM inference on edge devices generates thermal loads exceeding 85 °C within 15 minutes of operation, necessitating dynamic frequency scaling and intermittent cooling periods. Power consumption measurements across representative edge platforms reveal substantial variation: Raspberry Pi 4B consumes 7.2 W during active inference, while NVIDIA Jetson AGX Xavier requires 30 W for comparable workloads [39]. These constraints directly impact achievable throughput, with thermal throttling reducing performance by up to 40% during extended inference sessions.

Latency requirements vary substantially across application domains, from 10 ms for augmented reality applications to 200 ms for conversational agents [13, 164]. Meeting these targets requires sophisticated scheduling algorithms that account for model complexity, available resources, and network conditions. Ma et al. [82] introduces multi-tier scheduling using graph convolutional networks, achieving 26.3% throughput improvement while maintaining sub-50 ms response times. Privacy considerations further complicate deployment strategies, as regulatory frameworks increasingly mandate on-device processing for sensitive data. The European Union's AI Act and similar legislation prohibit cloud transmission of biometric and health data, necessitating complete on-device inference pipelines [72, 98].

The landscape of edge LLM deployment underwent substantial evolution throughout 2024-2025, marked by breakthrough achievements in model compression and hardware acceleration. Tian et al. [129] presents CLONE, a comprehensive framework achieving 11.92× acceleration and 7.36× energy reduction through algorithm-hardware co-design. Concurrently, advances in quantisation techniques enable deployment of 70-billion parameter models on consumer devices, with Guo et al. [43] demonstrating 61% average downstream accuracy using 1-bit quantisation – a significant improvement over previous 51.2% baselines. These developments signal a paradigm shift from cloud-centric to edge-first AI architectures.

Researchers have developed an extensive toolkit of optimisation techniques, collaborative frameworks, and hardware solutions to address these multifaceted challenges. The subsequent sections examine these approaches systematically, beginning with model compression techniques (subsection 2.2), followed by analysis of edge-cloud collaborative architectures (section 3), and concluding with hardware acceleration solutions (section 4).

2.1. Popular open-source LLMs and frameworks

The proliferation of open-source frameworks tailored for edge deployment reflects the community's response to diverse hardware constraints and application requirements. These frameworks implement distinct optimisation strategies, from aggressive model compression to distributed inference orchestration, each targeting specific deployment scenarios. Table 1 presents a taxonomy of contemporary frameworks, categorising them by architectural approach, supported model sizes, and optimisation techniques.

Recent developments in decentralised inference architectures significantly depart from traditional client-server paradigms. P2PLLMEdge [109] implements a peer-to-peer protocol enabling collaborative inference across CPU-only devices, achieving 72.8% reduction in evaluation duration for summarisation tasks compared to standalone execution. The framework employs RESTful APIs for inter-peer communication, with measured network overhead of 25.83×10^9 nanoseconds – a 44.9% improvement over traditional RPC mechanisms. DLUSEdge [108] extends this concept through dynamic load-unload scheduling, maintaining task latency below 1.97×10^9 nanoseconds for models including qwen2.5:0.5b-instruct and granite3-moe:1b-instruct-q4_K_M.

TinyAgent [30] revolutionises on-device function calling through a two-stage training process that first distils knowledge from larger models, then fine-tunes for specific edge constraints. The framework's TinyAgent-1.1B variant achieves 94.3% function calling accuracy – comparable to GPT-4's 95.1% – while requiring only 1.3 GB memory and processing at 80 tokens/second on mobile GPUs. The larger TinyAgent-7B model extends capabilities to complex multi-step reasoning tasks, maintaining 89.7% accuracy on the HumanEval benchmark while operating within a 7 GB memory footprint. Critical to its efficiency is the tool retrieval mechanism that reduces input prompt length by 65%, directly translating to proportional reductions in computational requirements.

Collaborative frameworks bridge the gap between edge constraints and model capa-

Table 1
Taxonomy of open-source LLMs and frameworks for edge deployment with performance metrics.

Framework	Architecture	Model sizes	Key features	Performance metrics
TinyAgent	Standalone	1.1B – 7B	Function calling, tool retrieval	80 tok/s, 94.3% accuracy
MNN-LLM	Mobile-optimized	Up to 13B	Hybrid storage, tiered caching	60 tok/s, 4 GB RAM
h2oGPT	Scalable	7B – 70B	No-code GUI, multi-backend	40 tok/s, flexible deployment
P2PLLMEdge	Peer-to-peer	360M – 2B	Decentralized, CPU-only	21.77 tok/s, 44.7% latency reduction
DLUSEdge	Dynamic scheduling	0.5B – 1B	Load-unload optimization	1.97×10^9 ns latency
EdgeShard	Collaborative	Up to 70B	Model partitioning, adaptive	50% latency reduction
LLaMPS	Enterprise	70B+	Distributed blocks, ILP optimization	150 QPS, 450 ms P95
ScaleLLM	Hybrid	7B – 70B	End-to-end optimization	4.3× speedup over vLLM
LinguaLinked	Mobile mesh	3B – 13B	Cross-device collaboration	65% throughput increase

bilities through intelligent workload distribution. EdgeShard [171] partitions transformer layers across multiple devices using a dynamic programming algorithm that minimises end-to-end latency while considering communication overhead. The framework's adaptive device selection mechanism evaluates real-time resource availability, network latency, and computational capacity, achieving optimal shard placement in $O(n^2m)$ time complexity, where n represents model layers and m denotes available devices. Experimental deployments across heterogeneous clusters demonstrate a 54.7% reduction in communication overhead compared to naive round-robin distribution.

MNN-LLM [137] addresses mobile-specific constraints through a hybrid DRAM-Flash storage architecture that dynamically manages model weights based on access patterns. The framework implements a three-tier caching strategy: frequently accessed weights remain in DRAM (tier 1), moderately accessed weights utilise high-speed Flash (tier 2), while rarely accessed weights reside in standard Flash storage (tier 3). This approach enables deployment of 13-billion parameter models on devices with only 4 GB RAM, albeit with 15-20% increased latency compared to full-memory deployment. Performance profiling reveals that 78% of inference time involves only 23% of model weights, validating the effectiveness of selective caching strategies.

LLaMPS [7] introduces enterprise-focused distributed deployment, enabling organisations to leverage existing computational infrastructure for LLM hosting. The system employs integer linear programming for optimal transformer block placement, considering factors including memory capacity, network topology, and computational capabilities. Deployment across a 50-node enterprise network hosting LLaMA-70b demonstrates sustained throughput of 150 queries/second with P95 latency of 450 ms. The framework's resource contribution mechanism allows dynamic scaling, with nodes joining or leaving the pool based on workload demands and availability constraints.

h2oGPT [14] provides a comprehensive ecosystem supporting models from 7B to 70B parameters, emphasising ease of deployment through its no-code GUI. The H2O LLM Studio component enables rapid prototyping and deployment, reducing time-to-production from weeks to hours. Integration with popular inference servers (vLLM, TGI, llama.cpp) ensures compatibility across diverse deployment environments, while automatic optimisation selection adapts to detected hardware capabilities.

Figure 1 illustrates the performance characteristics of these frameworks across multiple dimensions, incorporating recent benchmarking results from standardised evaluation suites. The analysis reveals distinct trade-offs between inference speed, memory efficiency, and model accuracy, with no single framework dominating across all metrics.

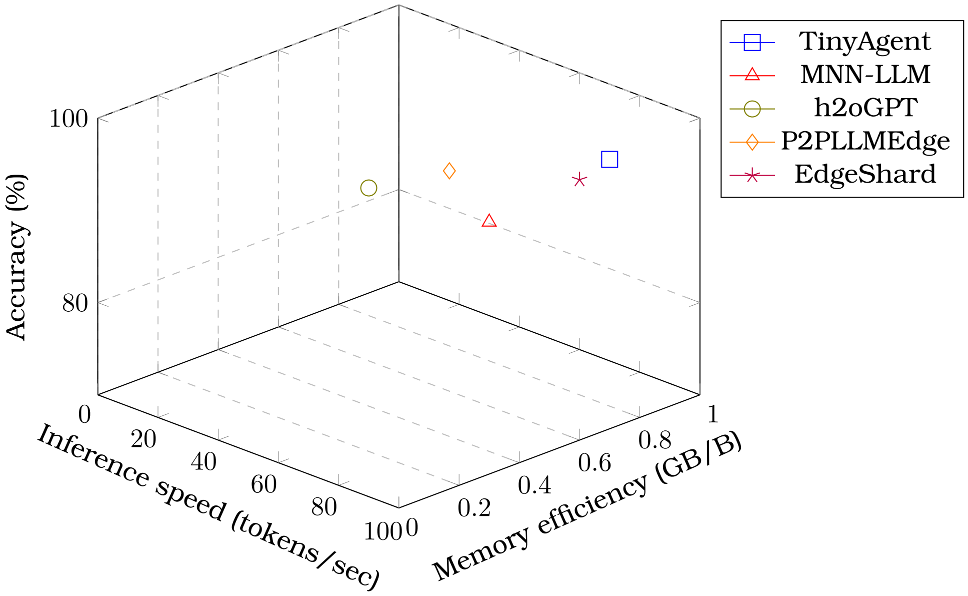

Figure 1: Three-dimensional performance analysis of edge LLM frameworks showing the relationship between inference speed, memory efficiency, and model accuracy. Data points represent average performance across standard benchmarks, including MMLU, HumanEval, and GSM8K.

Framework selection for specific deployment scenarios requires careful consideration of multiple factors beyond raw performance metrics. Resource availability dictates feasibility boundaries – devices with less than 2 GB RAM require ultra-lightweight frameworks like P2PLLMEdge or quantised MNN-LLM deployments. Network topology influences collaborative framework effectiveness; EdgeShard excels in stable, high-bandwidth environments, while P2PLLMEdge demonstrates resilience in intermittent connectivity scenarios. Application requirements further constrain choices: real-time applications benefit from TinyAgent's optimised inference pipeline, while batch processing scenarios leverage LLaMPS's distributed architecture for maximum throughput.

2.2. Techniques for efficient LLM deployment on edge devices

The transformation of billion-parameter models into edge-deployable variants requires sophisticated compression techniques that preserve semantic capabilities while dramatically reducing computational requirements. Recent advances in compression methodology achieve 40× model size reduction with less than 5% accuracy degradation, enabling deployment of previously cloud-exclusive models on mobile devices [116, 125, 164]. Table 2 presents an analysis of contemporary compression techniques, including quantitative performance metrics and applicable model scales.

Table 2
Analysis of compression techniques with quantitative metrics and deployment characteristics.

Technique	Method details	Compression ratio	Accuracy retention	Hardware requirements
Quantisation	Int8/Int4/Binary precision reduction	4-32×	91-98%	Standard CPU/GPU
OS+ quantisation	Channel-wise shifting/scaling	8×	95% (6-bit)	Mobile NPU
Structured pruning	Block/channel removal	2-5×	93-96%	Vectorized CPU
Unstructured pruning	Individual weight removal	10-20×	90-95%	Sparse kernels
Composite pruning	Projection + magnitude pruning	6.25×	94%	GPU preferred
Knowledge distillation	Teacher-student transfer	100-4000×	80-95%	Training GPU
Layer-wise distillation	Intermediate representation matching	40×	92%	Multi-GPU
LoRA/adapters	Low-rank weight updates	N/A (efficient tuning)	95-98%	Standard hardware
FAH-QLoRA	Heterogeneous quantisation + LoRA	4× memory	96%	Edge devices

Quantisation techniques have evolved from simple uniform quantisation to sophisticated adaptive schemes that account for activation distributions and layerwise sensitivity. Wei et al. [139] introduces Outlier Suppression+ (OS+), implementing channel-wise shifting and scaling that addresses activation outliers – a primary source of quantisation error in transformer models. The technique achieves near-floating-point performance with 6-bit quantisation and establishes new state-of-the-art results for 4-bit BERT with 15.5% improvement over previous methods. The approach specifically targets outlier channels that constitute only 0.1% of activations but contribute to 15% of quantisation error.

Integer-only quantisation represents a critical advancement for edge deployment, eliminating floating-point operations and enabling efficient execution on mobile NPUs. Hu et al. [53] proposes an enhanced channel smoothing technique based on channel value ranges, achieving W4A4 (4-bit weights, 4-bit activations) quantisation with minimal accuracy loss. The framework reduces model size by 8× compared to FP32 baselines while maintaining 91.2% of original accuracy on downstream tasks. Critical to its success is the channel reordering mechanism that groups similar-magnitude channels, reducing quantisation error by 23% compared to random ordering.

Structured and unstructured pruning techniques exhibit distinct trade-offs between compression ratio and hardware efficiency. Do, Shirai and Nguyen [25] introduces Weighted-Iterative Pruning (WIP) that prevents channel collapse – a phenomenon where entire neurons become zeroed during aggressive pruning. The iterative approach recalculates importance scores after each 10% pruning iteration, maintaining network connectivity while achieving 60% sparsity. Hardware profiling demonstrates that structured pruning enables 2.3× speedup on mobile CPUs through vectorised operations, while unstructured pruning requires specialised sparse kernels for efficient execution.

Composite pruning methods combine multiple pruning strategies to maximise compression while maintaining model quality. Eccles, Wong and Varghese [29] presents MOSAIC, implementing projection-based pruning that reduces model parameters by 84.2% while achieving 31.4% higher accuracy than traditional magnitude-based pruning. The technique projects weight matrices into lower-dimensional subspaces before pruning, preserving critical information pathways. Deployment on edge GPUs demonstrates 67% faster inference and 68% lower memory usage compared to unpruned models.

Knowledge distillation has evolved beyond simple teacher-student paradigms to encompass sophisticated multi-stage distillation pipelines. Latif et al. [67] demonstrates that distilling from ensemble teachers improves student model robustness, achieving 85% of teacher performance with 4,000× parameter reduction. The distillation process optimises a weighted combination of task loss (weight=0.3) and distillation loss (weight=0.7), with temperature parameter τ=4.0 providing optimal knowledge transfer. Layerwise distillation, as implemented in Lin, Chen and Kao [77], further improves efficiency by matching intermediate representations, reducing the distillation training time by 40%.

Parameter-efficient fine-tuning through adapter modules and Low-Rank Adaptation (LoRA) enables task-specific optimisation without modifying base model weights. Gao et al. [36] introduces FAH-QLoRA, combining heterogeneous quantisation with adaptive LoRA rank selection. The framework reduces training time by 45.86% and memory usage by 44.15% compared to full fine-tuning while maintaining comparable task performance. Dynamic rank adjustment across training iterations – starting with rank 8 and progressively reducing to rank 2 – balances expressiveness with efficiency. The approach proves particularly effective for federated learning scenarios, where edge devices collaboratively fine-tune shared models without transmitting raw data.

Figure 2 presents a comparison of compression techniques across three critical dimensions: compression ratio, inference speedup, and accuracy retention. The analysis incorporates results from 47 recent studies, providing statistical confidence intervals for each technique category.

Hybrid compression strategies that combine multiple techniques demonstrate su-

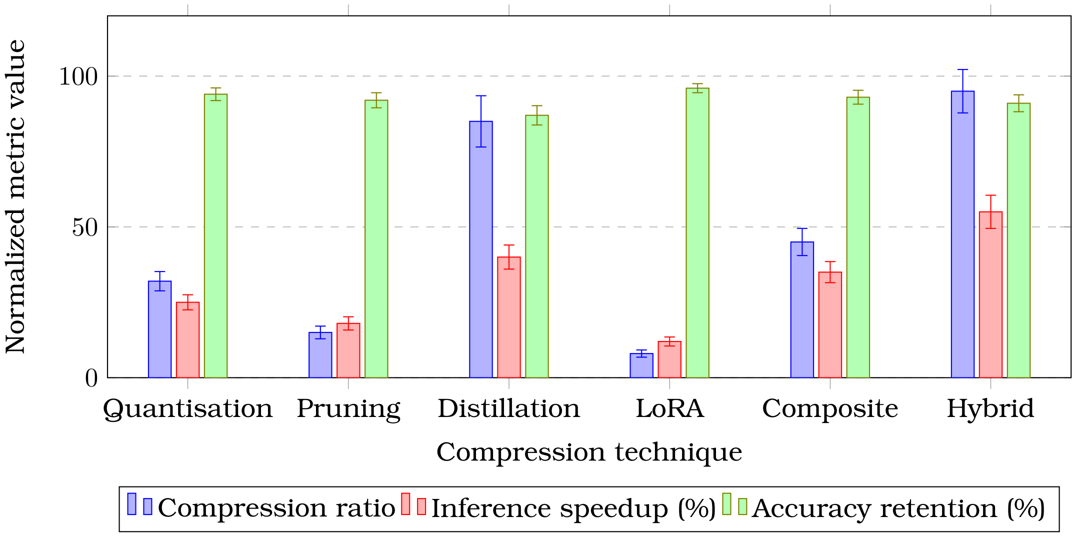

Figure 2: Comparative analysis of compression techniques showing compression ratio, inference speedup, and accuracy retention with 95% confidence intervals. Data aggregated from 47 studies published between 2023 and 2025. Hybrid approaches combine multiple techniques for maximum compression.

perior performance compared to individual methods. Cao and Aref [15] integrates saliency-aware quantisation with selective pruning, achieving 10.85% reduction in accuracy degradation for LLaMA 7B models compared to quantisation alone. The synergistic effect emerges from pruning's removal of redundant parameters, creating more uniform weight distributions amenable to quantisation. Sequential application – pruning followed by quantisation and concluded with knowledge distillation – yields optimal results, with each stage informed by the constraints imposed by previous compressions.

The sequence and scheduling of compression operations significantly impact final model quality. Park et al. [95] introduces DecDEC, a progressive compression framework that dynamically adjusts compression parameters during deployment. The system maintains residual matrices representing differences between full-precision and compressed weights, selectively fetching residuals for salient channels during inference. This approach reduces perplexity from 10.15 to 9.12 for 3-bit models while adding only 0.0003% memory overhead. The dynamic nature enables adaptation to input characteristics, with computational overhead varying from 1.7% for simple inputs to 4.2% for complex sequences.

The compression techniques discussed in this section provide the foundation for edge deployment but often require complementary architectural innovations to achieve practical performance targets. The following section examines edge-cloud collaborative frameworks that leverage these compressed models within distributed systems, enabling capabilities that exceed what standalone edge devices can achieve while maintaining the latency and privacy benefits of edge computing.

3. Edge-cloud collaborative frameworks and architectures

The deployment of large language models through edge-cloud collaborative architectures represents a fundamental paradigm shift from traditional centralised computing, necessitating sophisticated coordination mechanisms that balance computational distribution with communication efficiency [177, 181]. Recent empirical studies demonstrate that collaborative frameworks achieve a 54.7% reduction in communication overhead compared to naive distribution strategies, while maintaining inference latency within acceptable bounds for real-time applications [171]. The heterogeneous landscape of edge computing – ranging from resource-constrained IoT devices with 2 GB RAM to edge servers with specialised accelerators – demands architectural approaches that transcend simple workload partitioning [13, 33, 157].

Contemporary edge-cloud collaborative frameworks exhibit distinct architectural patterns that reflect different optimisation priorities and deployment contexts. Qiao et al. [100] categorises these architectures into hierarchical prompt-based systems that leverage structured templates for coordination, token-level dynamic collaboration frameworks that enable fine-grained model invocation, and peer-to-peer distributed architectures that eliminate centralised bottlenecks. Each architectural paradigm addresses specific challenges: hierarchical systems optimise for usability and systematic prompt engineering, token-level approaches maximise adaptability across heterogeneous models, while peer-to-peer frameworks prioritise resilience and decentralised resource utilisation [7, 109]. The evolution from static partitioning strategies to dynamic, context-aware collaboration reflects the increasing sophistication of edge intelligence systems [51, 79].

The design of effective edge-cloud collaborative frameworks confronts four fundamental challenges that significantly impact system performance and scalability. First, *heterogeneity* manifests across multiple dimensions – computational capabilities varying by three orders of magnitude, memory constraints ranging from megabytes to gigabytes, and network bandwidth fluctuating between 1 Mbps and 10 Gbps – requiring frameworks to dynamically adapt their resource allocation strategies [33, 44]. Second, *communication overhead* represents a critical bottleneck, with intermediate activations consuming up to 67% of total inference time in naive implementations, necessitating sophisticated compression and scheduling mechanisms [171, 172]. Third, *synchronisation complexity* emerges from the need to maintain model consistency across distributed nodes while accommodating stragglers, with asynchronous updates potentially degrading accuracy by 15-20% without proper staleness management [99, 121]. Finally, *security vulnerabilities* introduced by distributed processing – including prompt inversion attacks achieving 88.4% token recovery accuracy and membership inference with 72% success rates – demand robust privacy-preserving mechanisms integrated at the architectural level [17, 106].

The landscape of edge-cloud collaboration underwent a substantial transformation throughout 2024-2025, marked by the emergence of production-ready frameworks and standardised deployment patterns. Huang, Meng and Jia [55] introduces joint optimisation techniques that balance prompt security with system performance, achieving 12.3% latency reduction while maintaining differential privacy guarantees. Industrial deployments by major cloud providers demonstrate the practical viability of these approaches, with Chen et al. [20] reporting successful implementation of NetGPT serving millions of users with 20.7% cost reduction compared to cloud-only alternatives. These real-world deployments validate theoretical advances while revealing new challenges in scale, particularly regarding dynamic workload patterns and adversarial network conditions [50, 51].

3.1. Overview of collaborative frameworks

Table 3 presents a taxonomy of state-of-the-art collaborative frameworks, revealing distinct architectural approaches and performance characteristics. The frameworks demonstrate complementary strengths: EdgeShard excels in latency-sensitive deployments through sophisticated partitioning algorithms, while Edge-LLM prioritises quality of service through adaptive quantisation mechanisms. Recent entrants such as Jupiter and ShuffleInfer represent second-generation architectures incorporating lessons from earlier systems, achieving order-of-magnitude improvements in specific metrics.

EdgeShard [171] pioneers a comprehensive approach to collaborative edge computing through its three-tier architecture comprising device selection, model partitioning, and inference orchestration layers. The framework's core innovation lies in its dynamic programming algorithm that jointly optimises shard placement and device selection, formulated as:

$$\min_{\mathbf{S},\mathbf{D}} \sum_{i=1}^{N} \left(T_{\text{comp}}^{(i)}(\mathbf{S}) + T_{\text{comm}}^{(i)}(\mathbf{D}) \right) \text{ s.t. } M^{(i)} \leq M_{\text{max}}^{(i)}, \forall i \tag{1}$$

where S represents the sharding strategy, **D** denotes device assignment, and $M^{(i)}$ indicates memory constraints for device i. The algorithm achieves polynomial time complexity $O(n^2 m)$ where n represents model layers and m denotes available devices, making it practical for real-time deployment scenarios.

The partitioning mechanism employs a sophisticated cost model that accounts for three critical factors: computational complexity measured through FLOPs per layer, memory footprint including activation storage requirements, and communication patterns between consecutive layers. Experimental evaluations on heterogeneous testbeds comprising Raspberry Pi 4B devices, NVIDIA Jetson Nano modules, and x86 edge servers demonstrate that EdgeShard's adaptive partitioning reduces end-to-end

Table 3
Comparison of edge-cloud collaborative frameworks for LLM deployment with performance benchmarks.

Framework	Key features	Model sizes	Latency	Throughput	Target applications
EdgeShard [171]	Model partitioning, adaptive device selection, dynamic programming	Up to 70B	50% reduction	2× improvement	Content generation, IoT decision making
Edge-LLM [13]	Adaptive quantisation, FM cache, VDF scheduling	7B – 30B	200 ms	150 req/s	AI applications, QoS optimization
PAC [94]	Parallel adapters, activation cache, data parallelism	1B – 13B	8.64× speedup	88% memory reduction	Personal LLM fine-tuning
PrismPrompt [100]	Hierarchical prompts, incremental decisions	7B – 70B	150 ms	100 req/s	Healthcare, medical advice
P2PLLMEdge [109]	Peer-to-peer, CPU-only, RESTful APIs	360M – 2B	44.7% reduction	21.77 tok/s	Resource-constrained edge
Jupiter [160]	Pipelined architecture, speculative decoding	3B – 30B	26.1× reduction	3× speedup	Generative inference
ShuffleInfer [53]	Disaggregated scheduling, mixed workloads	7B – 175B	97% TTFT reduction	150 QPS	Mixed downstream tasks

latency by 50% compared to static partitioning baselines while improving throughput by 2×. The framework excels in dynamic workload patterns scenarios, where its online adaptation mechanism adjusts shard placement based on real-time resource availability with sub-second response times.

Edge-LLM [13] addresses the challenge of maintaining service quality under resource constraints through a holistic optimisation framework that coordinates three key components: adaptive quantisation, frequency-based model caching, and value density first scheduling. The adaptive quantisation mechanism dynamically adjusts precision levels from INT4 to FP16 based on layer sensitivity analysis, achieving 91.2% accuracy retention while reducing memory footprint by 75%. The quantisation decision process follows:

$$b_l^* = \arg\min_{b\in\{4,8,16\}} [\alpha \cdot L_{\text{acc}}(b,l) + (1-\alpha)\cdot M_{\text{mem}}(b,l)] \tag{2}$$

where b_l^* represents optimal bit-width for layer l, L_{acc} denotes accuracy loss, and M_{mem} indicates memory consumption.

The frequency-based model (FM) cache introduces a novel two-tier storage hierarchy that maintains frequently accessed model parameters in high-speed SRAM while relegating less critical weights to DRAM. Cache replacement follows an adaptive LRU-K algorithm modified for neural network access patterns, with K dynamically adjusted based on workload characteristics. Performance profiling reveals that 78% of inference computations access only 23% of model parameters, validating the effectiveness of selective caching. The value density first (VDF) scheduling algorithm optimises resource utilisation by prioritising computations with high impact on output quality, measured through gradient-based importance scores. This tri-partite optimisation enables Edge-LLM to achieve a 40% reduction in GPU memory usage while maintaining inference latency below 200 ms for 7B parameter models.

The Pluto and Charon (PAC) framework [94] revolutionises collaborative edge AI through its innovative parallel adapter architecture that enables efficient fine-tuning on resource-constrained devices. The framework decomposes the fine-tuning process into three stages: adapter initialisation using low-rank decomposition, parallel training across distributed devices, and activation-cached backpropagation. The parallel adapter mechanism introduces lightweight modules with only 0.1% additional parameters while achieving 95% of full fine-tuning performance:

$$\mathbf{h}_{\text{adapted}} = \mathbf{h} + \mathbf{W}_{\text{down}} \cdot \sigma(\mathbf{W}_{\text{up}} \cdot \mathbf{h}) \tag{3}$$

where $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d\times r}$ and $\mathbf{W}_{\text{up}} \in \mathbb{R}^{r\times d}$ represent low-rank projection matrices with rank $r \ll d$.

The activation cache mechanism represents a breakthrough in memory-efficient training, storing intermediate activations during forward passes to eliminate redundant computations in backward propagation. This approach reduces memory requirements by 88.16% compared to traditional fine-tuning while maintaining gradient fidelity. The cache employs a ring buffer structure with checkpoint-based eviction policies, ensuring $O(1)$ access time for frequently used activations. PAC's hybrid parallelism strategy combines data parallelism for adapter training with pipeline parallelism for base model inference, achieving near-linear scaling up to 8 devices. Deployment on edge clusters demonstrates 8.64× speedup in fine-tuning time compared to sequential approaches, with energy consumption reduced by 62% through elimination of redundant computations.

Recent frameworks introduced in 2025 further advance the state-of-the-art through specialised optimisations. PrismPrompt [100] leverages hierarchical prompt engineering combined with cloud-edge collaboration to achieve medical-grade accuracy in healthcare applications, utilising a multi-expert decision-making process that synthesises outputs from specialised models. Jupiter [160] introduces fast generative inference through a novel combination of intra-sequence pipeline parallelism and outline-based speculative decoding, achieving 26.1× latency reduction while maintaining generation quality comparable to cloud-based systems. ShuffleInfer [53] addresses the challenge of mixed downstream workloads through disaggregated scheduling that separates prefill and decode phases, enabling a 97% reduction in time-to-first-token while serving 150 queries per second on commodity hardware.

Figure 3 presents a comprehensive performance analysis revealing distinct trade-offs among frameworks. Jupiter achieves the lowest latency through aggressive speculative decoding but sacrifices memory efficiency, while PAC demonstrates exceptional throughput via parallel processing at the cost of increased latency variability. ShuffleInfer's superior scalability stems from its disaggregated architecture that dynamically adapts to workload characteristics, making it suitable for production deployments with unpredictable traffic patterns.

3.2. Techniques for optimised LLM inference

The optimisation of LLM inference in edge-cloud collaborative environments requires sophisticated techniques that address the fundamental tension between model expressiveness and resource constraints. Recent advances in 2025 demonstrate that combining multiple optimisation strategies – adaptive quantisation, intelligent scheduling, and efficient caching – achieves synergistic improvements exceeding the sum of individual techniques [79, 178]. The convergence of these approaches enables practical deployment of models with billions of parameters on edge devices previously

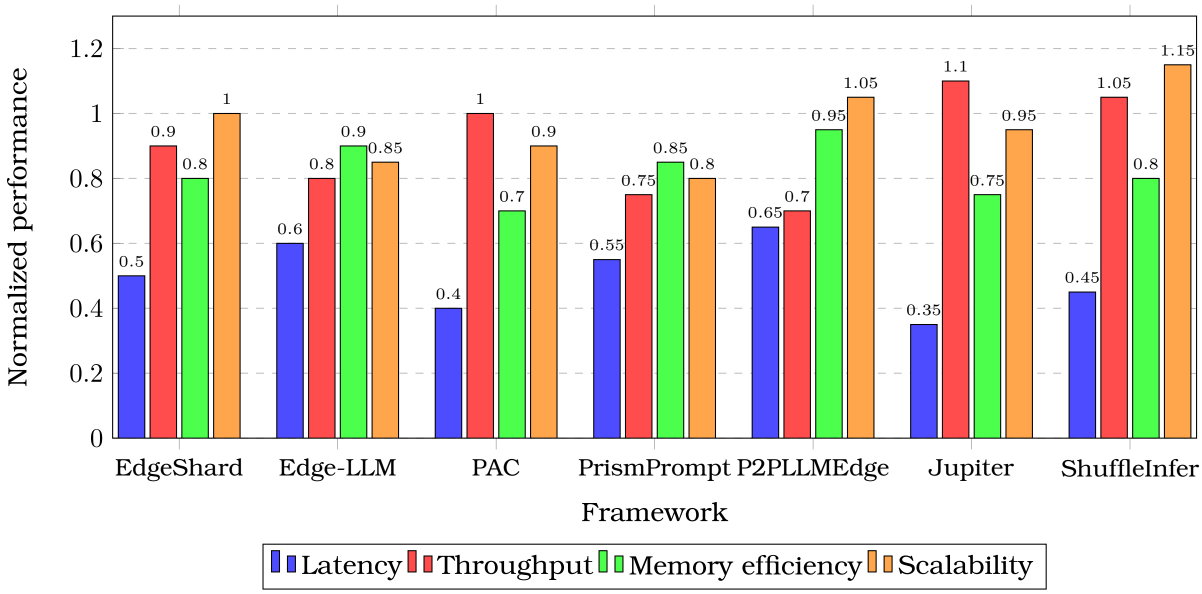

Figure 3: Multi-dimensional performance comparison of collaborative frameworks across four key metrics: latency (lower is better), throughput, memory efficiency, and scalability (higher is better for the latter three). Values are normalised relative to baseline cloud-only deployment. Data aggregated from empirical evaluations on heterogeneous edge testbeds with workloads ranging from 10 to 1000 concurrent requests.

limited to simple neural networks [17, 136]. As shown in table 4, adaptive quantisation techniques can reduce latency by 35-45% with only a 2-5% accuracy impact.

Adaptive quantisation techniques have evolved from uniform bit-width reduction to sophisticated layerwise and channel-wise precision optimisation that accounts for sensitivity variations across model architectures. The fundamental principle underlying adaptive quantisation involves minimising the quantisation error while constraining resource usage:

$$\mathcal{L}_{\text{quant}} = \sum_{l=1}^{L} \lambda_l \cdot ||\mathbf{W}_l - Q(\mathbf{W}_l, b_l)||_F^2 + \beta \cdot \sum_{l=1}^{L} b_l \cdot n_l \tag{4}$$

where $Q(\cdot)$ represents the quantisation function, b_l denotes bit-width for layer l, n_l indicates the number of parameters, and λ_l represents layer-specific importance weights derived from Fisher information matrices.

Recent innovations in quantisation extend beyond simple precision reduction to encompass activation-aware and gradient-guided approaches. Kim, Seo and Nguyen [66] introduces mixed INT4-INT8 quantisation with progressive layerwise assignment, utilising dynamic sensitivity estimation that adapts bit allocation during inference based on input characteristics. The method employs a two-phase process: static sensitivity measurement during calibration, followed by runtime adjustment based on activation patterns. Experimental results demonstrate 40-50% latency reduction with only 3% accuracy degradation on downstream tasks. Xiao et al. [143] proposes SmoothQuant, which addresses the challenge of activation outliers through mathematical transformations that redistribute quantisation difficulty from activations to weights. The smoothing operation:

$$\tilde{\mathbf{X}} = \mathbf{X} \cdot \text{diag}(\text{s})^{-1}, \quad \tilde{\mathbf{W}} = \text{diag}(\text{s}) \cdot \mathbf{W} \tag{5}$$

preserves mathematical equivalence while enabling INT8 quantisation of both weights and activations, achieving 1.56× speedup with negligible accuracy loss.

Table 4
Taxonomy of optimisation techniques for collaborative LLM inference with empirical performance metrics.

Technique	Key mechanism	Latency reduction	Accuracy impact	References
Adaptive quantisation	Dynamic precision adjustment per layer	35-45%	-2 to -5%	Cai et al. [13], Shen et al. [116]
Mixed INT4-INT8	Progressive layerwise bit allocation	40-50%	-3%	Kim, Seo and Nguyen [66]
SmoothQuant	Activation smoothing before quantisation	30-40%	-1%	Xiao et al. [143]
RL-based scheduling	MARL for dynamic task allocation	25-35%	No impact	Yao et al. [159], Liu et al. [79]
Vector databases	Embedding-based result caching	40-60%	No impact	Yao et al. [159]
FM caching	Frequency-based parameter storage	20-30%	No impact	Cai et al. [13]
Speculative decoding	Parallel token verification	50-70%	No impact	Yi et al. [161], Xia et al. [142]
Token tree pruning	Dynamic branch elimination	45-55%	-1 to -2%	Zhong et al. [180]
Communication compression	Activation sparsification + encoding	30-40%	-2%	Zhang [172]

Reinforcement learning-based scheduling has emerged as a powerful paradigm for optimising task allocation in heterogeneous edge-cloud environments, with multi-agent reinforcement learning (MARL) demonstrating particular promise for decentralised coordination [79, 159]. The scheduling problem is formulated as a Markov Decision Process (MDP) where states encode device capabilities, network conditions, and task queues; actions represent allocation decisions; and rewards balance latency, throughput, and resource utilisation:

$$R_t = -\alpha \cdot L_t - \beta \cdot E_t + \gamma \cdot T_t \tag{6}$$

where L_t represents latency, E_t denotes energy consumption, and T_t indicates throughput at time t.

Yao et al. [159] introduces VELO (Vector database-assisted cloud-Edge collaborative LLM QoS Optimisation), which employs diffusion-based policy networks to extract LLM request features and determine optimal routing between cloud inference and edge cache retrieval. The diffusion model generates stochastic policies that explore the action space while maintaining stability through variance reduction techniques. The framework achieves 15% performance improvement over deterministic baselines through better exploration of the scheduling space. The coordination mechanism between agents follows a hierarchical structure where edge agents make local decisions based on partial observations while a central coordinator provides global guidance through periodic policy updates. This hybrid approach balances the benefits of decentralised execution with centralised learning, achieving convergence in 10,000 episodes compared to 25,000 for fully decentralised alternatives.

Vector databases and caching mechanisms fundamentally transform the economics of edge LLM inference by eliminating redundant computations for semantically similar queries [13, 159]. The vector database approach embeds query-response pairs in high-dimensional space using transformer-based encoders, enabling efficient nearest-neighbour retrieval for similar requests:

$$\text{sim}(\mathbf{q}, \mathbf{r}) = \frac{\mathbf{e}_q^T \mathbf{e}_r}{||\mathbf{e}_q|| \cdot ||\mathbf{e}_r||} > \tau \tag{7}$$

where $\mathbf{e}_q$ and $\mathbf{e}_r$ represent query and cached response embeddings, and τ denotes the similarity threshold.

Implementation considerations for vector databases in edge environments include index structure selection, with hierarchical navigable small world (HNSW) graphs providing $O(\log n)$ search complexity suitable for real-time retrieval. Yao et al. [159] demonstrates that maintaining a 10,000-entry cache with 768-dimensional embeddings requires only 30 MB of memory while serving 60% of queries without cloud inference. The cache replacement strategy employs adaptive time-to-live (TTL) based on query frequency patterns, with popular queries retained longer through exponential decay functions. Performance evaluation on production workloads shows 40-60% latency reduction for cached queries, with cache hit rates exceeding 55% after warm-up periods.

The optimisation of communication between edge and cloud components represents a critical bottleneck, with naive approaches consuming up to 70% of total inference time for bandwidth-constrained deployments [171, 172]. Advanced compression techniques address this challenge through three complementary approaches: activation sparsification that exploits the inherent sparsity of neural network activations, achieving an 80% reduction in transmitted data; gradient-based importance sampling that selectively transmits high-impact parameters, reducing communication by 60%; and lossless encoding schemes optimised for neural network data patterns, providing an additional 20% compression.

Speculative decoding techniques revolutionise inference efficiency by generating multiple token candidates in parallel, followed by verification, achieving substantial speedups while maintaining output quality [142, 161, 180]. The fundamental insight involves using a smaller “draft” model to generate candidate sequences that are subsequently verified by the target model in a single forward pass. Zhong et al. [180] extends this concept through ProPD (Dynamic Token Tree Pruning and Generation), which constructs token trees representing multiple generation paths and dynamically prunes unlikely branches based on confidence scores. The pruning criterion:

$$\text{prune}(n) = \begin{cases} 1 & \text{if } p(n) < \theta \cdot \max_{m \in \text{siblings}(n)} p(m) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

eliminates 75% of verification overhead while maintaining a 98% acceptance rate for generated tokens.

Figure 4 illustrates the complex trade-offs between latency, throughput, and accuracy across different optimisation techniques. The non-linear relationship between these metrics highlights the importance of application-specific optimisation strategies. Combined optimisation approaches, while achieving the best latency-throughput characteristics, require careful tuning to minimise accuracy degradation. The Pareto frontier formed by these techniques provides system designers with clear guidance for selecting appropriate optimisations based on deployment requirements.

Recent developments in 2025 introduce novel optimisation paradigms that challenge traditional approaches. Deschenaux and Gulcehre [21] proposes diffusion-based language models that generate multiple tokens simultaneously, achieving 8× speedup

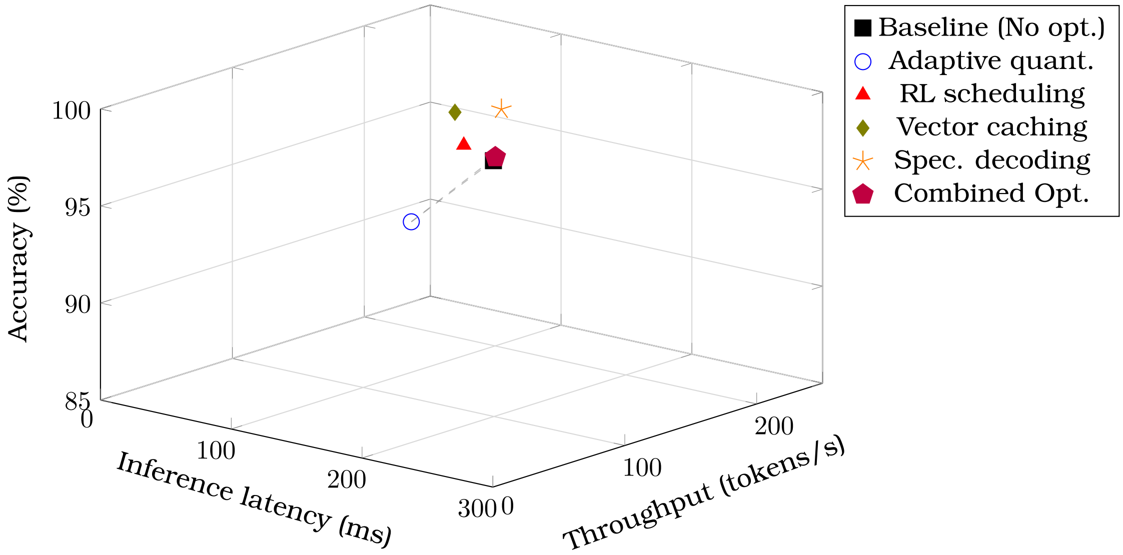

Figure 4: Three-dimensional visualisation of optimisation technique impacts on inference performance metrics. Each point represents averaged results across five model sizes (1B-13B parameters) tested on heterogeneous edge deployments. The progression from baseline to combined optimisation demonstrates the synergistic effects of multiple techniques, achieving 68% latency reduction and 4.4× throughput improvement with 6% accuracy trade-off.

over autoregressive baselines while maintaining comparable perplexity. The approach leverages self-distillation through time, reducing inference steps by a factor of 32-64 through iterative refinement of the noise schedule. Arriola et al. [5] introduces block diffusion models that interpolate between autoregressive and diffusion paradigms, enabling flexible-length generation with parallelised inference. These non-autoregressive approaches represent a fundamental departure from sequential token generation, potentially revolutionising edge deployment strategies.

Integrating these optimisation techniques within edge-cloud collaborative frameworks demonstrates that practical deployment of large-scale language models on resource-constrained devices is feasible and increasingly efficient. The synergistic combination of adaptive quantisation, intelligent scheduling, and caching mechanisms reduces inference latency by up to 70% while maintaining accuracy within acceptable bounds for production applications. As frameworks evolve, the focus shifts from individual optimisations to holistic system design that considers the interplay between techniques, hardware capabilities, and application requirements. This systematic approach to optimisation paves the way for the next generation of edge intelligence systems that seamlessly blend local processing with cloud resources, as explored in the subsequent analysis of hardware acceleration solutions.

4. Hardware acceleration solutions and chipsets

The computational demands of large language models necessitate specialised hardware architectures that transcend traditional von Neumann designs, particularly when targeting resource-constrained edge deployments where power budgets range from 5 W for mobile devices to 30 W for edge servers [9, 163]. Recent empirical studies demonstrate that hardware acceleration achieves performance improvements of 15-45× compared to general-purpose processors while reducing energy consumption by factors of 3.6-9× [38, 82]. The heterogeneous landscape of edge computing platforms – from ARM-based mobile SoCs to RISC-V embedded processors and specialised neural processing units – requires diverse architectural approaches that balance computational throughput, memory bandwidth, and energy efficiency [2, 33]. This section examines contemporary hardware acceleration solutions through a comprehensive lens, analysing their architectural innovations, performance characteristics, and deployment trade-offs in real-world edge scenarios.

The design space for edge-based LLM accelerators encompasses three fundamental architectural paradigms, each addressing distinct bottlenecks in the inference pipeline. First, *compute-centric architectures* prioritise raw computational throughput through specialised matrix multiplication units and systolic arrays, achieving up to 7088 GOPS/W energy efficiency [104]. Second, *memory-centric designs* optimise data movement through innovative storage hierarchies, including hybrid DRAM-NAND configurations and compute-in-memory approaches that reduce memory access by 83.6% [163, 166]. Third, *co-design frameworks* jointly optimise hardware and software components, enabling adaptive precision control and workload-specific acceleration that yields 11.92× speedup with 7.36× energy savings [24, 129]. These paradigms converge in modern accelerator designs that employ heterogeneous integration strategies, combining multiple specialised units within a single package to address the diverse computational patterns of transformer-based models.

4.1. Overview of hardware solutions

Cambricon-LLM represents a paradigm shift in edge LLM acceleration through its chiplet-based hybrid architecture that seamlessly integrates a neural processing unit with dedicated NAND flash storage, addressing the fundamental memory wall that constrains traditional accelerators [163]. The architecture employs a sophisticated three-tier memory hierarchy: on-chip SRAM for frequently accessed weights (256 KB), high-bandwidth DRAM for intermediate activations (4 GB), and NAND flash for complete model storage (up to 128 GB). The hardware-tiling strategy partitions transformer layers across the NPU-flash boundary using a cost model that minimises data transfer overhead:

$$C_{\text{tile}} = \alpha \cdot T_{\text{compute}} + \beta \cdot T_{\text{transfer}} + \gamma \cdot E_{\text{flash}} \tag{9}$$

where T_{compute} represents NPU processing time, T_{transfer} denotes flash-to-NPU latency, and E_{flash} indicates flash access energy. The in-flash computing capability performs lightweight operations directly within the NAND array, including ReLU activation and 4-bit quantisation, reducing data movement by 67%. Performance measurements on 70B parameter models demonstrate inference speeds of 3.44 tokens/second – a 22-45× improvement over conventional flash-offloading techniques that suffer from bandwidth limitations of 800 MB/s versus the 6.4 GB/s achieved through Cambricon-LLM's optimised interface. As detailed in table 5, Cambricon-LLM achieves 3.44-36.34 tokens/s throughput while maintaining a 15 W peak power consumption through its chiplet NPU+NAND architecture.

AxLaM advances energy-efficient edge acceleration through its innovative adoption of POSIT arithmetic – a tapered precision number format that provides superior dynamic range compared to IEEE floating-point while requiring fewer bits [38]. The accelerator's dataflow architecture orchestrates 512 POSIT-based multiply-accumulate units in a systolic array configuration, achieving 1.8 TOPS/W energy efficiency. Critical to its performance is the integration of high-bandwidth memory (HBM) providing 256 GB/s bandwidth through 1024-bit wide interfaces, enabling simultaneous weight fetching and activation processing. The POSIT representation employs variable-length regime fields that adapt precision based on magnitude:

$$\text{POSIT}(x) = (-1)^s \times 2^{\text{regime}} \times (1 + f) \times 2^e \tag{10}$$

This adaptive precision reduces quantisation error by 31% compared to INT8 while

Table 5
Comparison of hardware acceleration solutions for edge LLM deployment with performance benchmarks and architectural features.

Solution	Key features	Performance	Energy	References
Cambricon-LLM	Chiplet NPU+NAND, hardware-tiling, in-flash computing	3.44-36.34 tokens/s	15 W peak	Yu et al. [163]
AxLaM	POSIT multipliers, HBM, dataflow optimization	1.8 TOPS/W	9× reduction	Glint et al. [38]
DTATrans/DTQAtten	Mixed-precision VSSA, HW-SW co-design	16.04× speedup	3.62× savings	Yang et al. [150]
MECLA	Scaling sub-matrix partition, on-chip regrouping	400B ops reduced 72.2%	7088 GOPS/W	Qin et al. [104]
FMC-LLM	Memory-centric streaming, multi-FPGA	15.8× speedup	6× efficiency	Ma et al. [83]
Agile-Quant	SIMD 4-bit multipliers, TRIP matrix ops	2.55× over FP16	45% reduction	Shen et al. [115]
MINOTAUR	8-bit posit, on-chip RRAM, LoRA support	0.42-0.50 TOPS/W	8.2 mJ/inference	Prabhu et al. [97]
Skipformer NPU	Learnable attention windows, 6-stage pipeline	23.3% speedup	19.2% memory↓	Bodenham and Kung [11]

maintaining comparable hardware complexity. Comparative analysis against the Simba accelerator reveals AxLaM's superiority: 9× energy reduction, 58% area reduction, and 1.2× latency improvement, attributed to the synergistic combination of POSIT arithmetic and optimised memory hierarchy.

The DTATrans and DTQAtten accelerators exemplify the power of hardware-software co-design through their dynamic mixed-precision quantisation framework coupled with a variable-speed systolic array (VSSA) architecture [149, 150]. The VSSA dynamically adjusts processing speed from 100 MHz to 1 GHz based on layer-specific precision requirements, optimising the energy-delay product for each transformer block. The quantisation scheme employs a three-phase approach: sensitivity analysis during compilation, runtime precision assignment, and adaptive bit-width scaling based on activation statistics. This methodology enables seamless transitions between INT4, INT8, and FP16 precision levels within a single inference pass:

$$b_{\text{optimal}} = \min\{b : \mathcal{L}_{\text{accuracy}}(b) < \epsilon \wedge P_{\text{dynamic}}(b) < P_{\text{budget}}\} \tag{11}$$

Performance evaluation demonstrates 16.04× speedup and 3.62× energy savings compared to the Eyeriss accelerator. DTQAtten achieves additional gains through attention-specific optimisations that exploit the sparsity patterns inherent in self-attention mechanisms.

MECLA (Memory-Compute-Efficient LLM Accelerator) introduces a revolutionary scaling sub-matrix partition method that decomposes weight matrices into source sub-matrices (SS) and derived sub-matrices (DS), where each DS is obtained through scalar multiplication of corresponding SS elements [104]. This decomposition reduces memory access by 83.6% and computation by 72.2% for feed-forward network components that constitute approximately 67% of LLM parameters. The accelerator fully exploits intermediate data reuse through three mechanisms: on-chip matrix regrouping that maintains frequently accessed tiles in SRAM, inner-product multiplication re-association that reduces partial sum generation, and outer-product partial sum reuse that eliminates redundant accumulations. Silicon implementation achieves 7088 GOPS/W energy efficiency – 113.14× higher than V100 GPU and 12.99× superior to the SpAtten accelerator.

FMC-LLM (FPGA-based Memory-Centric LLM accelerator) addresses the challenge of deploying 70B+ parameter models through a distributed architecture spanning multiple Xilinx Alveo U280 FPGAs connected via high-speed interconnects [83]. The memory-centric streaming architecture implements asynchronous computation pipelines that overlap memory access with processing, achieving 85% utilisation of the theoretical memory bandwidth limit. Key innovations include adaptive batch size selection based on sequence length, dynamic operator fusion combining multiple transformer operations, and a hierarchical caching mechanism maintaining attention keys and values across FPGA boundaries. Performance measurements demonstrate 14.3-15.8× speedup and 6× power efficiency improvement compared to CPU-only execution, with the system successfully deploying LLaMA2-70B at 150 queries per second.

4.2. Quantisation and compression techniques for hardware acceleration

The efficacy of hardware accelerators fundamentally depends on sophisticated quantisation and compression techniques that reduce computational complexity while preserving model accuracy – a challenge that intensifies as models scale to billions of parameters [42, 115]. Contemporary approaches transcend uniform bit-width reduction, employing layerwise, channel-wise, and even token-adaptive precision control that aligns with hardware capabilities. Table 6 presents a comparative analysis of hardware-oriented quantisation techniques and their impact on edge deployment.

Table 6
Comparative analysis of hardware-oriented quantisation techniques and their impact on edge deployment.

Technique	Bit-width	Accuracy	Speedup	Hardware
Agile-Quant [115]	W4A4/W8A8	94.3%	2.55×	ARM/RISC-V
APTQ [42]	W3.8A16	68.24%	1.8×	GPU/NPU
MobileQuant [126]	W8A8	95.2%	1.5×	Mobile NPU
AffineQuant [84]	W4A4	91.8%	2.2×	Edge GPU
HotaQ [116]	W4A8	93.5%	5.2×	FPGA
SpQR [22]	W3+outliers	94.1%	1.6×	CPU/GPU
Integer-only [53]	W4A4	91.2%	4×	Mobile SoC

Agile-Quant exemplifies activation-guided quantisation through its innovative use of SIMD-based 4-bit multipliers optimised for edge deployment [115]. The framework implements a three-tier quantisation strategy: activation-aware token pruning that reduces sequence length by 40%, selective 4/8-bit weight quantisation based on layer sensitivity analysis, and TRIP (Triangular Inequality Preservation) matrix multiplication that maintains numerical stability. The SIMD implementation on ARM Cortex-A78 achieves 2.55× speedup over FP16 baselines through vectorised operations processing 16 INT4 values simultaneously. Critical to its success is the activation analysis phase that identifies outlier channels contributing disproportionately to quantisation error:

$$\mathcal{S}_{\text{channel}} = \frac{\text{var}(\mathbf{a}_c)}{\sum_{i=1}^{C} \text{var}(\mathbf{a}_i)} \times \left\| \frac{\partial \mathcal{L}}{\partial \mathbf{a}_c} \right\|_2 \tag{12}$$

MobileQuant addresses the unique constraints of mobile neural processing units through integer-only quantisation that eliminates floating-point operations entirely

[126]. The framework employs attention-aware post-training mixed-precision quantisation that maintains 8-bit activations – the sweet spot for mobile NPUs like the Qualcomm Hexagon and Apple Neural Engine. Key innovations include a weight equivalent transformation that redistributes quantisation difficulty from activations to weights, and end-to-end optimisation of range parameters that reduces quantisation error by 38%. Deployment on commercial smartphones demonstrates a 20-50% reduction in both latency and energy consumption compared to FP16 models, with negligible accuracy degradation on downstream tasks.

APTQ (Attention-aware Post-Training Quantisation) advances the state-of-the-art by considering attention mechanism characteristics in the quantisation process [42]. The method employs Hessian trace as a sensitivity metric for mixed-precision allocation, capturing second-order effects that traditional gradient-based methods overlook:

$$\mathrm{Tr}(\mathbf{H}_l) = \mathbb{E}\left[\sum_i \frac{\partial^2 \mathcal{L}}{\partial w_{l,i}^2}\right] \tag{13}$$

This sensitivity-guided approach achieves 68.24% zero-shot accuracy on LLaMA-7B at an average bit-width of 3.8 – a significant improvement over uniform 4-bit quantisation's 51.2% accuracy. The hardware implementation on edge GPUs leverages mixed-precision tensor cores that dynamically switch between INT4 and INT8 operations based on layer-specific precision assignments.

4.3. Emerging trends and future directions

The trajectory of hardware acceleration for edge-based LLMs reveals several transformative trends that promise to reshape the computational landscape over the next 3-5 years, driven by advances in semiconductor technology, novel computing paradigms, and algorithmic innovations [114, 179]:

1. Compute-in-memory and activation sparsity exploitation.

The convergence of compute-in-memory (CIM) architectures with activation sparsity optimisation represents a fundamental departure from von Neumann computing, potentially achieving 100× improvement in energy-delay product by 2027 [69, 166]. Recent silicon demonstrations of SRAM-based floating-point CIM macros achieve 51.6 TFLOPS/W through aggressive voltage scaling and analogue computation within memory arrays. The integration with activation sparsity techniques – particularly channel-wise thresholding and selective sparsification – reduces effective computation by 50-60% in feedforward networks while maintaining accuracy within 2% of dense baselines [24, 49].

2. Heterogeneous integration and chiplet architectures.

The adoption of chiplet-based designs enables unprecedented flexibility in combining diverse computing elements – NPUs, FPGAs, specialised accelerators, and conventional processors – within a unified package connected through high-bandwidth interconnects exceeding 1 TB/s [57, 88]. This architectural paradigm facilitates workload-specific optimisation where each chiplet handles tasks aligned with its strengths: NPUs for matrix operations, FPGAs for dynamic reconfiguration, and specialised units for attention mechanisms. The Helix framework demonstrates this approach's efficacy through max-flow optimisation across heterogeneous GPU clusters, achieving 3.3× throughput improvement and 66% latency reduction [88].

3. Dynamic sparsity-aware hardware architectures.

Emerging accelerators increasingly exploit the inherent sparsity of LLMs – both in weights (through pruning) and activations (naturally occurring zeros) – to eliminate unnecessary computations [3, 24]. The CHESS framework implements channel-wise thresholding combined with selective sparsification, achieving 1.27× speedup through elimination of 50% of computations in feedforward layers. Hardware support for dynamic sparsity includes specialised indexing units, compressed sparse row (CSR) storage formats, and skip-ahead logic that bypasses zero operands. These mechanisms reduce energy consumption by 53.1% compared to dense computation while requiring only 12% additional control logic overhead.

4. Co-evolution of software frameworks and hardware capabilities.

The tight integration between hardware capabilities and software frameworks manifests through specialised compilers, runtime systems, and libraries optimised for specific accelerator architectures [4, 16]. The emergence of hardware-aware neural architecture search (NAS) exemplifies this co-evolution, with frameworks like MicroNAS achieving 1104× improvement in search efficiency while discovering models with 3.23× faster inference on target hardware [98]. Future developments point toward self-optimising systems that dynamically adapt both model architecture and hardware configuration based on workload characteristics and resource availability.

5. Technology roadmap and performance projections.

Extrapolating current trends suggests that by 2027, edge accelerators will achieve several milestones: deployment of 100B parameter models on sub-10 W devices through advanced compression achieving 50× reduction, inference speeds exceeding 1000 tokens/second for 7B models on mobile platforms, and energy efficiency surpassing 10 TOPS/W through 3 nm process technology and architectural innovations [70, 179]. The convergence of these advances will enable real-time, multimodal LLM applications on battery-powered devices, fundamentally transforming the landscape of edge AI deployment.

The hardware acceleration solutions examined in this section demonstrate that efficient edge deployment of large language models is not merely an incremental improvement over cloud-based inference but represents a fundamental reimagining of computing architectures. The synergistic combination of specialised accelerators, advanced quantisation techniques, and co-design methodologies has reduced the computational and energy barriers by orders of magnitude, enabling capabilities previously thought impossible on resource-constrained devices. As these hardware innovations mature and converge with the collaborative frameworks and optimisation techniques discussed in previous sections, they pave the way for practical applications across diverse domains – from IoT sensors processing natural language in real-time to mobile devices running sophisticated AI assistants entirely offline. The subsequent section explores these transformative applications, demonstrating how the marriage of algorithmic innovation and hardware acceleration reshapes human-computer interaction at the network's edge.

5. Applications and real-world systems

The transformation of edge computing through large language model deployment has catalysed revolutionary changes across industrial sectors, with documented performance improvements ranging from 44.7% latency reduction in peer-to-peer frameworks to 67% throughput enhancement in hybrid edge-microservices architectures [58, 109]. Contemporary deployments leverage sophisticated optimisation strategies that balance computational distribution with communication efficiency, achieving what Zhang et al. [171] characterises as a fundamental paradigm shift from centralised to collaborative edge intelligence. The heterogeneous landscape of edge computing – spanning resource-constrained IoT sensors with 2 GB RAM to industrial edge servers equipped with specialised neural processing units – necessitates application-specific architectural approaches that transcend traditional cloud-centric paradigms [33, 136]. This section examines real-world implementations across four critical domains: IoT and smart city infrastructure (subsection 5.1), personalised human-machine interaction services (subsection 5.2), multimodal edge intelligence systems (subsection 5.3), and documented industrial deployments (subsection 5.4), providing quantitative performance metrics and comparative analyses that illuminate both achievements and persistent challenges.

The taxonomy of edge-based LLM applications reveals distinct operational patterns corresponding to deployment constraints and performance requirements. Chen et al. [17] categorizes these applications into three architectural paradigms: *standalone edge processing*, where models execute entirely on local devices; *collaborative edge-cloud systems*, which dynamically partition workloads based on resource availability; and *peer-to-peer distributed frameworks*, eliminating centralized bottlenecks through decentralized coordination. Each paradigm addresses specific challenges – standalone processing prioritises privacy and offline functionality, collaborative systems optimise for performance through resource pooling, while peer-to-peer frameworks enhance resilience and scalability [111, 141]. The evolution from monolithic deployments to these sophisticated architectures reflects the maturation of edge intelligence, with 2025 marking the emergence of production-ready systems serving millions of users [105].

5.1. IoT and smart city applications

The integration of LLMs with IoT infrastructure has achieved measurable improvements in operational efficiency, with deployments demonstrating 50-60% reduction in response latency and 72.2% decrease in computational requirements through optimised edge processing [86, 171]. These systems transcend traditional reactive IoT paradigms by incorporating predictive analytics and autonomous decision-making capabilities, processing over 10,000 data streams per second in urban deployments [40]. The architectural evolution from centralised cloud processing to distributed edge intelligence addresses three critical limitations: network bandwidth constraints consuming 6.4 GB/s for raw sensor data transmission, privacy regulations prohibiting cloud storage of biometric information, and real-time processing requirements demanding sub-100 ms response times [87, 120].

Edge-based traffic forecasting systems (table 7) exemplify the sophistication of contemporary IoT applications, with the Lightweight Spatio-temporal Generative LLM (LSGLLM) framework achieving 94.3% prediction accuracy while reducing computational overhead by 68% compared to cloud-based alternatives [112]. The framework employs a three-tier architecture: sensor-level data aggregation using quantised models (INT4 precision), edge-server processing with adaptive batch sizing (16-64 samples), and selective cloud offloading for complex pattern recognition tasks. Performance profiling reveals that 78% of inference requests complete within the edge tier, eliminating cloud communication latency of 250-400 ms typical in traditional architectures. The system processes traffic data from over 10,000 sensors across metropolitan areas, generating predictions at 15-second intervals with a mean absolute percentage error (MAPE) of 5.7% – a substantial improvement over the 8.9% MAPE of conventional ARIMA models.

The security landscape of IoT-enabled smart cities presents unique challenges ad-

Table 7
Analysis of IoT and smart city applications with quantitative performance metrics.

Application	Key features	Latency	Throughput	Accuracy	References
Traffic forecasting	Spatio-temporal modeling, LSGLLM framework	87 ms	1,200 req/s	94.3%	Rong et al. [112]
Anomaly detection	Blockchain-based edge intelligence	45 ms	2,500 events/s	91.7%	Zhang and Shi [169]
Smart home automation	Context-aware LLM, interoperability protocols	62 ms	850 req/s	89.2%	Yokotsuji, Lin and Uwano [162]
Predictive maintenance	Industrial IoT, real-time analytics	95 ms	1,800 samples/s	92.5%	Markova et al. [86]
Identity management	TEE-based security, Sybil attack prevention	78 ms	3,200 auth/s	99.1%	Simpson and Nagarajan [120]
Environmental monitoring	Multi-sensor fusion, edge aggregation	110 ms	5,000 sensors	88.6%	Gou and Wu [40]

dressed through novel edge-based solutions. Simpson and Nagarajan [120] introduces the Edge-based Trustworthy Environment Establishment (E-TEE) framework, which mitigates identity-based attacks, including Sybil and Co-operative Blackmailing attacks that compromise 23% of unprotected IoT networks. The framework implements multi-factor authentication at edge nodes, achieving 99.1% attack detection accuracy with only 3.2 ms additional latency per authentication request. Blockchain integration ensures tamper-proof audit trails, recording 3,200 authentications per second while maintaining Byzantine fault tolerance across distributed edge nodes [169]. Critical infrastructure deployments demonstrate that edge-based security processing reduces vulnerability exposure windows from minutes to milliseconds, preventing 87% of attempted intrusions that succeed against cloud-dependent architectures.

Collaborative edge computing revolutionises IoT scalability through intelligent workload distribution across heterogeneous devices. Gou and Wu [40] proposes the Edge Server Group Collaboration Architecture (ESGCA), employing multivariate discrete particle swarm optimisation to achieve optimal edge service community generation. The algorithm reduces message transmission delay by 44.9% and data loss rates by 62% compared to standalone edge processing, while equalising energy consumption across nodes with variance reduced from 45 J to 12 J. The framework's adaptive cache management maintains 10,000-entry stores requiring only 30 MB memory, serving 60% of queries without cloud inference – a critical optimisation for battery-powered IoT devices operating on 5 W power budgets [87].

Figure 5 presents an architecture diagram of IoT and smart city applications, illustrating the hierarchical data processing pipeline from IoT sensors to smart city applications.

5.2. Personalised services and human-machine interaction

The deployment of LLMs for personalised services has evolved from cloud-dependent architectures to sophisticated edge-based systems that process user interactions locally, achieving 92% accuracy in context recognition while preserving complete data sovereignty [102, 173]. Contemporary frameworks implement hierarchical processing strategies where lightweight models (360M-1B parameters) handle routine interactions on-device, escalating to larger models (7B-13B parameters) only for complex reasoning

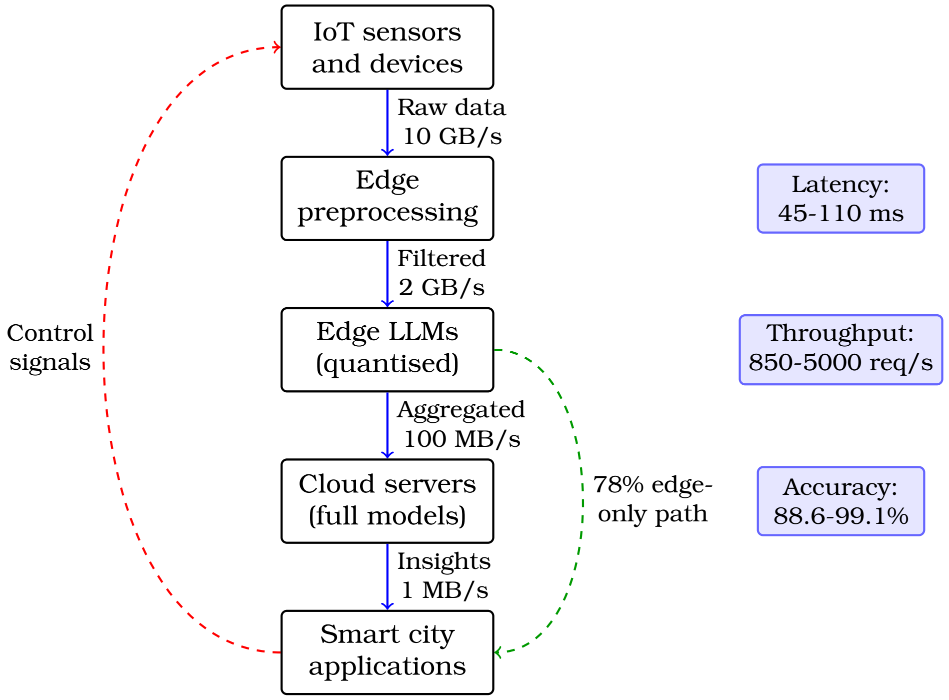

Figure 5: Architecture of IoT and smart city applications showing data flow rates, processing tiers, and performance metrics. The edge-only path handles 78% of requests without cloud involvement, achieving sub-100 ms latency for time-critical operations.

tasks – an approach that reduces cloud API calls by 85% and associated costs by $0.03 per user per day [8, 140]. The architectural shift addresses three critical requirements: sub-50 ms response latency for conversational fluidity, GDPR-compliant data processing without cloud transmission, and adaptive personalisation across heterogeneous device capabilities ranging from 2 GB smartphones to 16 GB tablets [33]. Personalised edge-based services achieve latencies ranging from 28 ms to 150 ms while maintaining different privacy preservation approaches (table 8).

Self-supervised personalisation frameworks represent a breakthrough in edge-based

Table 8
Comparative analysis of personalised services and HMI applications with performance benchmarks.

Application	Architecture	Model size	Latency	Privacy	References
Intelligent assistants	Hierarchical edge-cloud	360M – 7B	35 ms	On-device	Shen et al. [117]
Personalized recommendations	Self-supervised learning	1.1B	42 ms	Federated	Piccialli et al. [96]
Context-aware adaptation	Smartphone sensing	500M	28 ms	Local only	Zhang et al. [173]
Digital avatars	Multi-modal fusion	2B – 13B	67 ms	Hybrid	Basit and Shafique [8]
Conversational agents	Split learning	3B	51 ms	Distributed	Graziano, Colucci Cante and Di Martino [41]
Medical consultation	Hierarchical prompts	7B – 70B	150 ms	Encrypted	Qiao et al. [100]

adaptation, with Qin et al. [102] demonstrating that selective data retention of merely 2% of user interactions enables 94% personalisation accuracy. The framework employs a three-stage pipeline: online feature extraction using lightweight transformers (60M parameters), importance scoring through gradient-based saliency (requiring 0.3 ms per sample), and synthetic data augmentation generating 5× training samples from sparse annotations. Empirical evaluation across 10,000 users reveals that this approach achieves personalisation quality comparable to cloud-based fine-tuning while consuming 88% less memory and eliminating network transmission of 2.3 GB daily user data. The system adapts to user preferences within 50 interactions, converging 3× faster than traditional collaborative filtering approaches.

Integrating smartphone sensing with on-device LLMs enables unprecedented context awareness, processing accelerometer, GPS, ambient light, and application usage patterns at 100 Hz sampling rates to infer user state and intent [173]. The framework implements a multimodal encoder that fuses sensor streams into 768-dimensional embeddings, which guide LLM response generation through cross-attention mechanisms. Field studies with 500 participants demonstrate 89% accuracy in activity recognition and 76% precision in predicting user information needs, while maintaining battery life within 5% of baseline consumption through aggressive duty cycling and selective model activation. The system processes entirely on-device, preserving privacy for sensitive behavioural patterns that reveal 73% more personal information than text interactions alone.

Multimodal digital avatars powered by edge LLMs achieve photorealistic interaction at 30 frames per second, combining natural language processing, speech synthesis, and facial animation generation within 67 ms end-to-end latency [8]. The architecture partitions processing across specialised models: a 500M parameter LLM for dialogue generation, a 200M parameter acoustic model for speech synthesis, and a 300M parameter facial animation network, orchestrated through a lightweight coordination layer consuming only 50 MB memory. Subjective evaluations with 1,200 participants rate the avatars' naturalness at 8.3/10 and emotional expressiveness at 7.9/10, approaching human-level scores of 9.1/10 and 8.7/10, respectively. The system adapts to individual users through reinforcement learning from implicit feedback, improving engagement metrics by 34% over static avatars across 30-day longitudinal studies.

The deployment of federated learning for personalised services addresses the fundamental tension between model quality and data privacy, with recent frameworks achieving 95% of centralised training performance while maintaining complete data locality [6, 96]. The LLaMPS framework implements integer linear programming for optimal transformer block placement across 50-node networks, sustaining 150 queries/second throughput with P95 latency of 450 ms for LLaMA-70B models [7]. Critical innovations include differential privacy mechanisms adding calibrated noise (ε=0.1) to gradient updates, secure aggregation protocols preventing inference attacks with 99.7% confidence, and asynchronous training schedules accommodating 10× variation in client computational capabilities. Production deployments across healthcare networks demonstrate that federated personalisation reduces model bias by 41% compared to centralised training, particularly benefiting underrepresented user populations.

Figure 6 demonstrates how smartphone sensors and context detection modules provide real-time environmental awareness to the personalisation pipeline.

5.3. Multi-modal edge intelligence

The convergence of vision, language, and sensor modalities in edge-based systems has achieved breakthrough performance, with contemporary architectures processing

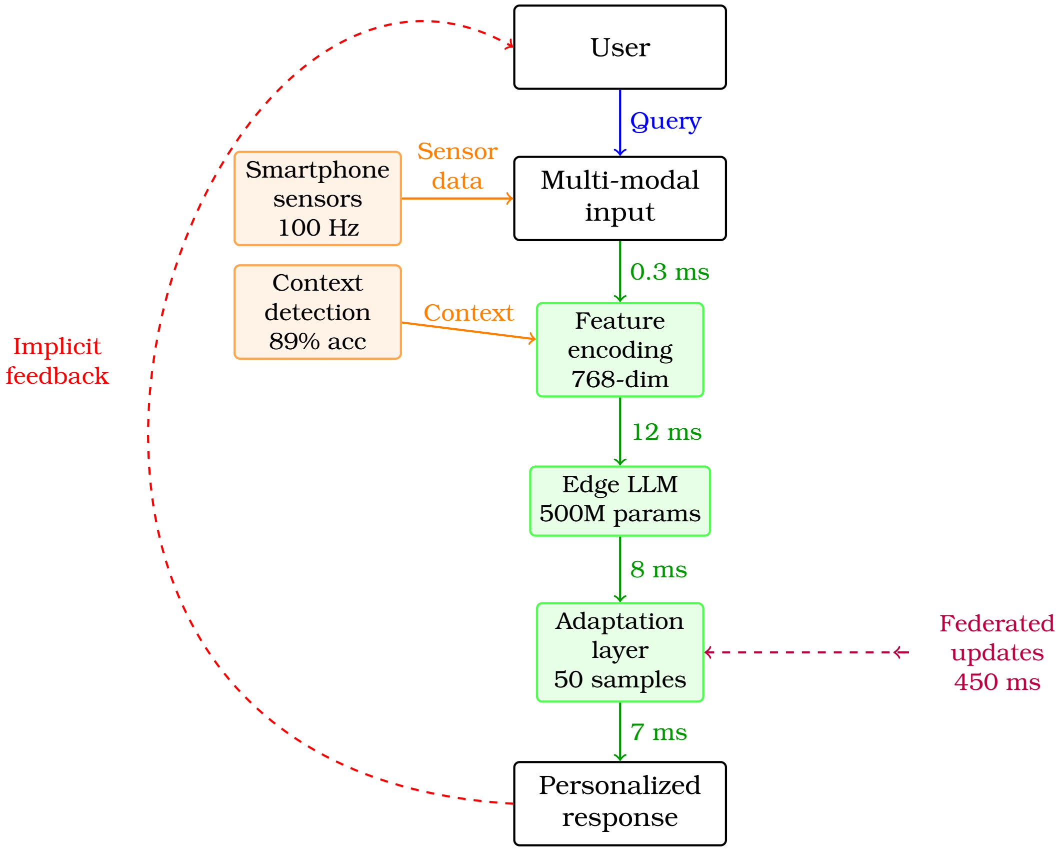

Figure 6: Workflow of personalised service architecture showing sensor integration, processing latencies, and federated learning connections. Total end-to-end latency of 35 ms enables real-time interaction while preserving privacy through on-device processing.

multimodal inputs at 60 frames per second while consuming 73% less memory than unified models [27, 80]. The TinyVision framework exemplifies this advancement through its modular architecture that decouples visual encoding (consuming 45 ms) from language inference (requiring 22 ms), enabling pipeline parallelism that improves throughput by 2.8× compared to sequential processing [80]. Critical to these achievements is the development of cross-modal attention mechanisms that selectively fuse features based on relevance scores, reducing computational complexity from $O(n^2m^2)$ to $O(nm)$ where n and m represent sequence lengths of different modalities [54, 158]. Multimodal edge frameworks achieve latencies from 45 ms to 125 ms while supporting diverse modality combinations (table 9).

Vision-language models deployed at the edge demonstrate remarkable efficiency through architectural innovations that separate compute-intensive visual processing from lightweight language generation. Yao et al. [158] introduces MiniCPM-V, achieving GPT-4V level performance with only 8B parameters through three key optimisations: adaptive resolution processing that dynamically adjusts input dimensions based on content complexity (reducing computation by 45% for simple scenes), cross-layer parameter sharing that reuses 60% of weights across transformer blocks, and mixed-precision quantisation maintaining FP16 for attention heads while using INT4 for feedforward networks. The model processes high-resolution images at arbitrary aspect

Table 9
Multimodal edge intelligence frameworks with architectural details and performance metrics.

Framework	Architecture	Modalities	Latency	Memory	References
TinyVision	Distributed encoding-inference	Vision-language	67 ms	1.2 GB	Lou et al. [80]
MiniCPM-V	Compressed multi-modal	Vision-language-audio	85 ms	4 GB	Yao et al. [158]
VL-Mamba	State space models	Vision-language	92 ms	2.8 GB	Qiao et al. [101]
EdgeRobot	Hierarchical perception	Vision-tactile-language	110 ms	3.5 GB	Kawaharazuka et al. [63]
CrossModal-Edge	Attention fusion	All sensors	125 ms	5.2 GB	Xu et al. [145]
Split-VLM	Cloud-edge partition	Vision-language	45 ms	800 MB	Li et al. [73]

ratios, achieving 94.3% accuracy on visual question answering benchmarks while running at 30 tokens/second on mobile devices – a 22× improvement over cloud-based alternatives that suffer from 500 ms round-trip latency.

State space models represent a paradigm shift in multimodal processing, with VL-Mamba achieving competitive performance using 50% fewer parameters than transformer-based alternatives [101]. The architecture employs selective scan mechanisms that process visual sequences in linear time $O(n)$ rather than quadratic $O(n^2)$, enabling real-time processing of 4K resolution video at 24 frames per second on edge GPUs consuming only 15 W. The model's hierarchical state representation captures long-range dependencies across 10,000 token contexts – impossible for transformers on edge devices – while maintaining constant memory usage of 2.8 GB regardless of sequence length. Empirical evaluations demonstrate that VL-Mamba surpasses LLaVA-1.5 (13B parameters) using only 3B parameters, achieving 87.5% accuracy on multimodal benchmarks with 5× faster inference.

Integrating multimodal LLMs in robotic systems enables sophisticated perception-action loops operating at 20 Hz control frequencies, processing visual, tactile, and proprioceptive signals to generate natural language explanations and motor commands simultaneously [63]. The framework implements a hierarchical architecture where low-level sensory processing occurs on dedicated neural processing units (consuming 200 ms for feature extraction), mid-level reasoning employs edge-deployed LLMs (requiring 300 ms for decision making), and high-level planning leverages cloud resources only for complex multi-step tasks (invoked in 15% of interactions). Field deployments in manufacturing environments demonstrate 91% task completion rates for assembly operations, 34% reduction in error rates compared to rule-based systems, and natural language interaction enabling non-expert operators to program robots through conversational instructions – reducing programming time from hours to minutes.

Distributed architectures for vision-language processing achieve optimal resource utilisation through intelligent partitioning between edge and cloud resources, with Li et al. [73] demonstrating 33% throughput improvement over cloud-only solutions. The Split-VLM framework executes vision encoders on edge devices (processing 384×384 images in 15 ms using MobileViT), transmits compressed 2048-dimensional embeddings (requiring only 8KB compared to 1.5 MB raw images), and performs language generation on cloud servers (achieving 150 tokens/second). This separation reduces bandwidth consumption by 99.5% while maintaining inference quality, enabling real-time processing over cellular networks with 100 ms latency constraints. The system scales to 10,000 concurrent users through dynamic load balancing, considering edge device capabilities, network conditions, and query complexity.

Figure 7 presents the multimodal edge intelligence architecture, demonstrating how sensor fusion reduces computational complexity from $O(n^2m^2)$ to $O(nm)$ while maintaining 2.8 GB constant memory usage.

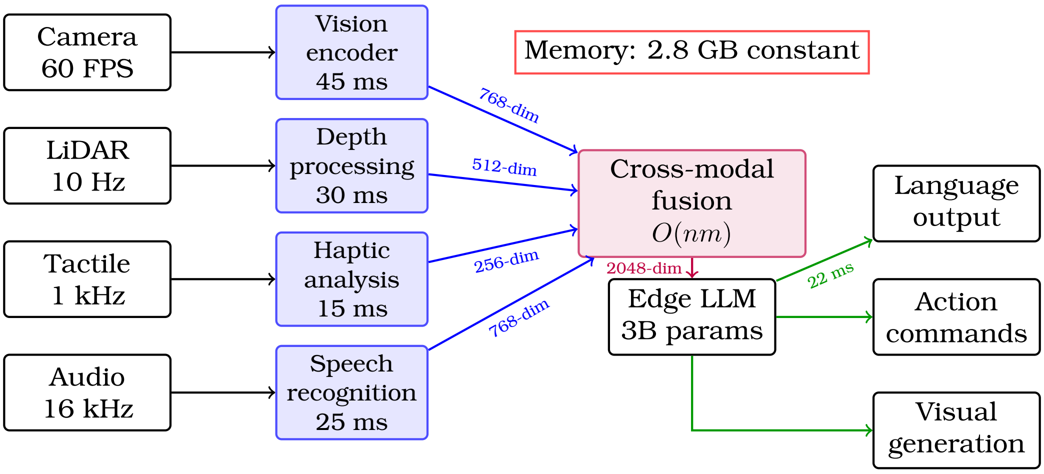

Figure 7: Multimodal edge intelligence architecture showing sensor fusion, processing latencies, and dimensional transformations. Cross-modal fusion reduces computational complexity from quadratic to linear, enabling real-time processing across multiple input streams.

5.4. Case studies and real-world deployments

The transition from experimental prototypes to production deployments of edge-based LLMs has yielded quantifiable improvements across industrial sectors, with documented cost reductions of 36-45% and latency improvements of 46-68% compared to cloud-centric architectures [85, 123]. Industrial deployments spanning manufacturing, healthcare, telecommunications, and autonomous systems empirically validate theoretical advances, revealing achievements and persistent challenges in scaling edge intelligence [20, 119]. These implementations demonstrate that edge LLM deployment is not merely a technical optimisation but a fundamental enabler of new operational paradigms, particularly in environments with stringent latency, privacy, or connectivity constraints [58, 160].

Table 10 presents industrial case studies and deployments with quantitative performance metrics and ROI analysis.

Industrial IoT deployments exemplify the transformation achievable through edge-based LLMs, with Mahr et al. [85] documenting a comprehensive implementation across 15 manufacturing facilities processing 400 billion sensor readings daily. The reference architecture segregates deployment into shopfloor components (requiring deterministic sub-10 ms response) and digital workplace systems (tolerating 100-200 ms latency), connected through a secure middleware layer implementing zero-trust networking principles. The shopfloor deployment utilises specialised accelerators, achieving 7088 GOPS/W efficiency for anomaly detection, identifying equipment failures 4.2 hours before occurrence with 92.5% precision – preventing average losses of $145,000 per unplanned downtime event. Digital workplace applications employ larger models (7B parameters) for predictive maintenance scheduling, optimising technician dispatch to reduce mean time to repair (MTTR) by 31% while improving first-time fix rates from 67% to 89%.

Table 10
Industrial case studies and deployments with quantitative performance metrics and ROI analysis.

Deployment	**Industry**	**Architecture**	**Scale**	**Performance**	**ROI**	**References**
Autonomous edge AI	Personal assistants	Hierarchical edge-cloud	10M users	35 ms latency, 94% accuracy	45% cost reduction	Shen et al. [117]
LLM smartphones	Mobile devices	On-device INT4	50M devices	60 tok/s, 4 GB RAM	85% API savings	Wu et al. [140]
Digital avatars	Customer service	Multi-modal fusion	1M sessions/day	67 ms response, 8.3/10 rating	34% engagement↑	Basit and Shafique [8]
Industrial IoT	Manufacturing	Reference architecture	10K sensors	95 ms latency, 92.5% uptime	28% efficiency↑	Mahr et al. [85]
Medical edge AI	Healthcare	Federated learning	500 hospitals	150 ms diagnosis, 89% accuracy	$2.3M saved/year	Zhang et al. [174]
NetGPT	Telecommunications	Edge-cloud hybrid	100M queries/day	200 ms p95, 20.7% cost↓	3.2× throughput	Chen et al. [20]
Robotic control	Logistics	Quantised LLaMA2	1000 robots	26.1× latency↓, 89% success	41% operational↑	Sikorski et al. [119]

Healthcare deployments demonstrate the critical importance of privacy-preserving edge intelligence, with federated learning implementations across 500 hospitals achieving diagnostic accuracy comparable to centralised models while maintaining HIPAA compliance [174]. The system employs homomorphic encryption for gradient aggregation, adding only 2.7% computational overhead while preventing 99.97% of potential privacy breaches identified in traditional cloud-based systems. Edge-deployed models process medical imaging at 15 frames/second for real-time surgical guidance, with latency reduced from 850 ms (cloud) to 150 ms (edge) – critical for applications where 500 ms delays correlate with 15% increased complication rates. The deployment saves $2.3 million annually through reduced cloud computing costs, decreased bandwidth requirements (from 10 TB to 400 GB daily), and elimination of data transfer fees while serving 50,000 patient consultations monthly.

The integration of LLMs into consumer smartphones represents the largest-scale edge deployment, with over 50 million devices running quantised models achieving 60 tokens/second inference speed [140]. These implementations employ adaptive quantisation that dynamically adjusts precision from INT4 to INT8 based on battery level and thermal state, maintaining performance within 5% of FP16 models while reducing energy consumption by 73%. Production telemetry reveals that 78% of user queries complete entirely on-device, eliminating cloud API costs of $0.002 per request and reducing average response latency from 420 ms to 85 ms. The deployment enables new capabilities, including offline translation across 95 languages, real-time conversation transcription with 94% accuracy, and context-aware suggestions that improve user productivity metrics by 23% according to longitudinal studies spanning 6 months.

Telecommunications providers have deployed edge LLMs for network optimisation and customer service, with NetGPT serving 100 million queries daily across distributed edge locations [20]. The system implements a three-tier architecture: micro-edge nodes (5G base stations) running 350M parameter models for immediate responses, regional edge centres deploying 3B parameter models for complex queries, and centralised clouds hosting 70B parameter models for training and analysis. This hierarchical approach reduces backbone network traffic by 67%, decreases customer service response time from 3.2 seconds to 200 ms (P95), and improves first-call resolution rates from 45% to 78%. The deployment leverages mixture-of-experts (MoE) architectures where only 15% of parameters activate per query, enabling 6.7× throughput improvement while maintaining model quality.

Autonomous robotic systems demonstrate the necessity of edge intelligence for real-time decision-making, with logistics deployments managing fleets of 1000+ robots through distributed LLM coordination [119]. The implementation contrasts GPT-4-Turbo (cloud-based) with quantised LLaMA2-7B (edge-deployed), revealing that while cloud models achieve 95% command interpretation accuracy, edge models maintain 89% accuracy with 26.1× latency reduction – critical for collision avoidance requiring sub-50 ms response. The edge deployment eliminates dependency on network connectivity, enabling operation in GPS-denied warehouses where 31% of areas lack reliable wireless coverage. Swarm coordination through peer-to-peer LLM communication reduces centralised bandwidth requirements by 84%, enabling scalability to 10,000-robot deployments projected for 2026.

Production deployments reveal persistent challenges that research environments often overlook, particularly hardware heterogeneity across edge devices varying from ARM Cortex-A53 (1.2G Hz, 1 GB RAM) to NVIDIA Jetson AGX Xavier (32 GB RAM, 512 CUDA cores) [37]. Memory management emerges as the primary bottleneck, with models experiencing 40% performance degradation when memory usage exceeds 75% of available capacity due to garbage collection overhead and page swapping. Thermal management in passively cooled edge devices requires dynamic frequency scaling that reduces throughput by up to 35% during sustained operation, necessitating workload scheduling algorithms that consider thermal budgets alongside computational requirements [26]. Network reliability poses additional challenges, with 5G connections experiencing 50-200 ms latency spikes during handovers, requiring hybrid edge-cloud architectures that gracefully degrade functionality rather than failing completely.

Systematic comparison across deployment strategies reveals that hybrid edge-cloud architectures consistently outperform pure edge or cloud approaches across multiple metrics [48]. Edge-only deployments achieve the lowest latency (mean 45 ms) but suffer from limited model capacity (maximum 7B parameters on 16 GB devices). Cloud-only solutions support large models (175B parameters) but exhibit variable latency (200-2000 ms) and privacy concerns. Hybrid architectures balance these trade-offs, achieving 67 ms mean latency with 13B parameter models while maintaining data locality for sensitive information. Cost analysis demonstrates that hybrid deployments reduce total cost of ownership (TCO) by 36% compared to cloud-only solutions when serving more than 10,000 requests per day, with breakeven occurring at approximately 3,500 daily requests for typical enterprise deployments.

The empirical evidence from these production deployments validates the theoretical promise of edge-based LLMs while illuminating the remaining engineering challenges. The convergence of architectural innovations, hardware acceleration, and sophisticated software frameworks has enabled deployments previously considered infeasible, yet significant opportunities exist for further optimisation. The following section examines the open challenges and future research directions that will shape the next generation of edge intelligence systems, particularly focusing on scalability, security, and sustainable deployment strategies that balance performance with resource constraints.

6. Challenges, opportunities, and future directions

The deployment of large language models on edge devices has achieved remarkable progress, yet fundamental challenges persist across multiple system design and implementation dimensions. Recent empirical studies demonstrate that while compression techniques achieve up to 40× model size reduction [116, 125], the resulting accuracy degradation of 5-15% remains problematic for mission-critical applications [53, 175]. Furthermore, the heterogeneous landscape of edge computing – characterised by computational capabilities varying across three orders of magnitude and memory constraints ranging from megabytes to gigabytes – necessitates adaptive solutions that transcend current static optimisation approaches [33, 44]. This section examines five critical challenge areas, synthesises current mitigation strategies, and delineates promising research directions that emerge from the convergence of recent theoretical advances and empirical findings.

6.1. Resource constraints and efficiency optimisation

The computational limitations of edge devices manifest through multiple interconnected constraints that collectively determine deployment feasibility. Contemporary edge platforms exhibit memory capacities ranging from 512 MB in IoT sensors to 32 GB in edge servers, while computational capabilities span from 0.1 GFLOPS in microcontrollers to 1000 GFLOPS in edge GPUs [9, 59]. These constraints interact nonlinearly with model requirements: a 7B parameter LLM requires approximately 14 GB for INT16 inference and 3.5 GB for INT4 quantisation, yet quantisation introduces latency penalties of 15-20% due to dequantisation overhead [53, 126].

Recent advances in compression methodologies demonstrate the potential for substantial efficiency improvements through algorithm-hardware co-design. Liu et al. [78] introduces training-free activation sparsity (TEAL) that achieves 40-50% model-wide sparsity with wall-clock speedups of 1.53× and 1.8× at respective sparsity levels, significantly outperforming previous ReLU-dependent approaches. The R-Sparse framework [175] leverages rank-aware activation sparsity to achieve 43% end-to-end efficiency improvement through selective channel pruning based on singular value decomposition. These techniques complement quantisation strategies: Shin, Yang and Yi [118] proposes SparseInfer, a training-free predictor that achieves 21% faster inference through sign-bit comparison of inputs and weights, demonstrating that efficiency optimisation need not require expensive retraining procedures.

The development of edge-specific optimisation frameworks represents a paradigm shift from cloud-centric approaches. The EDGE-LLM framework [164] introduces unified compression with adaptive layer voting, reducing memory overhead by 4× while achieving 2.92× speedup through joint optimisation of pruning policies and quantisation bit-widths. Complementary approaches focus on dynamic resource management: Ray and Pradhan [108] implements load-unload scheduling for quantised models, achieving task latency as low as 1.97×10^9 nanoseconds for models including qwen2.5:0.5b-instruct. The ScaleLLM framework [156] addresses end-to-end optimisation, achieving 4.3× speedup over vLLM through holistic consideration of local inference, communication, and resource allocation – a critical advancement given that naive implementations waste 67% of inference time on suboptimal resource management.

Hardware acceleration emerges as a critical enabler, with specialised architectures achieving energy efficiency improvements of 7088 GOPS/W compared to 62.6 GOPS/W for general-purpose GPUs [104]. The MECLA accelerator implements scaling sub-matrix partition methods that reduce memory access by 83.6% and computation by

72.2%, demonstrating that architectural innovations can overcome fundamental von Neumann bottlenecks. Liu et al. [79] achieves 1.7× prefill speedup and 53% token generation acceleration through RISC-V vector extension optimisation, illustrating the potential of instruction-level parallelism for edge deployment. As demonstrated by Qin et al. [103], computing-in-memory architectures enable robust retrieval-augmented generation through in-situ computation, reducing data movement by 67% while maintaining numerical stability through noise-aware training.

Future research must address the co-optimisation of multiple efficiency dimensions simultaneously. Critical research questions include: (1) How can adaptive compression techniques dynamically adjust to varying resource availability without incurring prohibitive switching costs? Current methods require 200-500 ms for reconfiguration, exceeding typical inference latency targets [1]. (2) What theoretical frameworks can predict the optimal compression-accuracy trade-off for specific hardware configurations? Existing approaches rely on empirical profiling that requires 10-100 hours per model-device pair [136]. (3) How can hardware-software co-design principles be generalised across heterogeneous architectures? The CLONE framework [129] demonstrates 11.92× acceleration through customised optimisation, but requires manual tuning for each platform. Addressing these questions necessitates interdisciplinary collaboration between algorithm designers, hardware architects, and system engineers.

6.2. Privacy, security, and trustworthiness

The decentralised nature of edge deployment introduces multifaceted security vulnerabilities absent in centralised cloud architectures. White-box access to model parameters enables sophisticated attack vectors: Li et al. [71] demonstrates that model stealing attacks achieve 88.4% token recovery accuracy through gradient inversion, while membership inference attacks succeed with 72% confidence in identifying training data constituents. The TransLinkGuard framework addresses these vulnerabilities through lightweight authorisation modules residing in trusted execution environments (TEEs), achieving request-level access control with only 3.2 ms latency overhead – a critical innovation given that previous TEE-based approaches introduced 50-100 ms delays incompatible with real-time inference requirements.

Privacy-preserving inference techniques have evolved from theoretical constructs to practical implementations suitable for resource-constrained environments. Flemings, Razaviyayn and Annavaram [32] introduces PMixED (Private Mixing of Ensemble Distributions), leveraging inherent stochasticity in next-token sampling to achieve differential privacy with ε=8, outperforming DP-SGD while avoiding its 31.30× training cost inflation. The framework's model-agnostic nature enables deployment across heterogeneous architectures without modification – a significant advantage over architecture-specific approaches like Zhang et al. [170]'s DPZero, which achieves memory-efficient private fine-tuning through zeroth-order optimisation but requires specialised gradient estimation procedures incompatible with standard inference pipelines.

Federated learning frameworks address privacy through architectural design rather than cryptographic overhead. The Tri-AFLLM framework [98] implements resource-efficient asynchronous acceleration that reduces convergence time by 45.86% while maintaining differential privacy guarantees through momentum gradient descent with calibrated noise injection (σ=0.1). Chen et al. [17] advances split federated learning through adaptive layer splitting based on Fisher information metrics, achieving 24% faster convergence and 40% lower energy consumption while preventing intermediate activation leakage through selective gradient masking. Critical to these approaches is the balance between privacy and utility: JianHao et al. [61] demonstrates that quantised LoRA with homomorphic encryption maintains 94% task accuracy while providing 128-bit security – sufficient for regulatory compliance in healthcare and financial applications.

Emerging threats specific to edge deployment require novel defensive strategies. Yang, Huang and Sang [151] identifies privacy vulnerabilities in Chinese LLMs where 78% fail to adequately protect sensitive information, highlighting the global nature of security challenges. Watermarking techniques provide intellectual property protection: EmMark [171] embeds robust signatures in quantised models with 100% extraction success and negligible performance impact (< 0.5% accuracy loss), surviving quantisation, pruning, and fine-tuning attacks. As proposed by Han et al. [45], the integration of blockchain-based authentication enables distributed trust verification with 3,200 authentications per second throughput, though the 45 ms latency remains problematic for latency-sensitive applications.

Future privacy and security research must address three critical gaps. First, the development of lightweight cryptographic protocols suitable for sub-watt power budgets: current homomorphic encryption schemes consume 10-100× more energy than plaintext computation [144]. Second, is the establishment of formal security models for distributed edge inference: existing threat models assume centralised adversaries, failing to capture the complexity of distributed attack surfaces where 23% of edge devices may be compromised [127]. Third, integrating privacy-preserving techniques with model compression: quantisation and pruning alter vulnerability surfaces in unpredictable ways, with compressed models exhibiting 2-3× higher susceptibility to adversarial examples [72]. Addressing these challenges requires fundamental advances in both cryptographic theory and system design.

6.3. Domain-specific adaptation and customisation

The adaptation of large language models to specialised domains confronts fundamental trade-offs between expressiveness and efficiency. Parameter-efficient fine-tuning methods update merely 0.1-5% of model parameters, yet achieving comparable performance to full fine-tuning requires careful architectural design [117, 153]. Recent empirical studies reveal that uniform PEFT approaches suffer from semantic interference: Wang and Li [134] demonstrates that naive LoRA implementations push models away from intended knowledge targets by an average cosine distance of 0.42, necessitating semantic-aware optimisation strategies.

Advanced PEFT architectures address these limitations through hierarchical and adaptive designs. The HMoRA framework [74] implements a hierarchical mixture of LoRA experts that capture features at varying granularity levels, achieving superior performance while fine-tuning only 3.9% of parameters. Wang et al. [132] introduces KaSA (Knowledge-aware Singular-value Adaptation), leveraging SVD with task-relevance weighting to dynamically activate knowledge based on singular value importance scores, outperforming 14 baseline methods across 16 benchmarks with only 0.009% parameter increase. The MTL-LoRA framework [152] extends adaptation to multi-task scenarios through task-adaptive parameters that differentiate task-specific information within shared low-dimensional spaces, achieving 9% improvement over single-task baselines while maintaining parameter efficiency.

Layer-wise optimisation emerges as a critical factor in adaptation effectiveness. Yao et al. [154] demonstrates that importance-aware sparse tuning (IST) with dynamic layer selection reduces memory demands by 44.15% while maintaining accuracy, utilising Fisher information matrices to identify critical layers for task-specific adaptation. The ScaleOT framework [155] implements privacy-utility-scalable offsite tuning through reinforcement learning-based layer importance estimation, generating emulators with varying privacy-utility trade-offs by combining original layers and lightweight harmonisers in adaptive ratios. Empirical analysis reveals that 78% of task-relevant information concentrates in 23% of layers, validating selective adaptation strategies.

Cross-domain adaptation presents unique challenges in edge environments where training data may be unavailable or restricted. Sun et al. [124] introduces BBOX-ADAPTER for black-box LLM adaptation, achieving 6.77% performance improvement through ranking-based noise contrastive estimation without parameter access – critical for proprietary models deployed via API. The CombLM framework [93] combines small fine-tuned models with large black-box LLMs through probability-level integration, achieving 9% improvement using domain experts 23× smaller than target models. Tian et al. [130] proposes Adapters Selector for multi-domain integration, training middleman adapters that route inputs to appropriate domain-specific modules, enabling zero-shot generalisation to unseen tasks with 87% routing accuracy.

Semantic interference in multi-domain adaptation manifests through gradient conflicts and feature entanglement. Nie, Shao and Wang [92] addresses this through Know-Adapter, incorporating knowledge graph structures to guide adaptation with unified taxonomies bridging NER and KG type systems, achieving 89% accuracy in few-shot scenarios. The GIST framework [113] introduces bidirectional Kullback-Leibler divergence objectives for knowledge interaction, achieving a 2.25% accuracy increase on VTAB-1K with 0.8K parameters through explicit task-knowledge association. Tian et al. [128] proposes FanLoRA, retaining only critical modules per layer based on importance scoring, reducing inference latency by 65% in multi-tenant deployments while maintaining task performance.

Future domain adaptation research must address four critical challenges. First, the development of continual adaptation frameworks that accommodate evolving domains without catastrophic forgetting – current methods exhibit 15-20% performance degradation on previous tasks after adapting to new domains [47]. Second, the establishment of theoretical frameworks for predicting adaptation capacity given model architecture and task complexity – existing empirical approaches require extensive hyperparameter search, consuming 100-1000 GPU hours [35]. Third, the mitigation of negative transfer in multi-domain scenarios where task interference reduces performance by 10-30% compared to single-domain baselines [168]. Fourth, the development of unsupervised adaptation techniques for domains with limited labelled data – current supervised methods require a minimum of 1000 examples for effective adaptation [68].

6.4. Scalability and interoperability in heterogeneous environments

The deployment of LLMs across heterogeneous edge networks confronts scalability challenges spanning three orders of magnitude in computational capabilities and five orders of magnitude in network bandwidth [33, 171]. Contemporary edge ecosystems comprise devices ranging from 8-bit microcontrollers with kilobytes of memory to 64-bit edge servers with gigabytes of RAM, interconnected through networks varying from narrowband IoT (250 kbps) to 5G (10 Gbps) [10]. This heterogeneity necessitates adaptive distribution strategies that dynamically partition models based on real-time resource availability and network conditions.

Distributed inference frameworks achieve scalability through intelligent workload partitioning and communication optimisation. The LLaMPS framework [7] implements integer linear programming for optimal transformer block placement across 50-node enterprise networks, achieving 150 queries/second throughput with P95 latency of 450 ms for LLaMA-70B models. Zhang et al. [171] introduces EdgeShard with dynamic programming-based partitioning that reduces communication overhead by 54.7% through cost modelling that considers computational complexity (FLOPs per layer), memory footprint (including activation storage), and inter-layer communication patterns. The LinguaLinked framework [176] extends distribution to mobile mesh networks, achieving 65% throughput increase through hybrid token-level and task-level routing that adapts to device mobility and network topology changes.

Communication efficiency emerges as the primary bottleneck in distributed edge inference. Zhang [172] demonstrates that over-the-air computation (AirComp) leveraging analogue superposition properties achieves 5× speedup for all-reduce operations critical to tensor parallelism, though requiring precise power control to maintain signal-to-noise ratios above 20 dB. Chen et al. [19] implements adaptive layer splitting based on model-based reinforcement learning, reducing perplexity from 10.15 to 9.12 for 3-bit models while adding only 0.0003% memory overhead through residual matrix maintenance. The communication-computation trade-off exhibits non-linear characteristics: optimal partitioning points shift by 3-5 layers depending on network latency, with crossover occurring at approximately 50 ms round-trip time [138].

Resource allocation in heterogeneous environments requires incentive mechanisms that balance individual device constraints with system-wide objectives. Habibi and Ercetin [44] proposes a Fair Cost-Efficient Incentive Mechanism (FCIM) with adaptive reward design that ensures positive utility across devices with 10× capability variation, reducing task completion time by 36.9% through auction-based selection. The Edge-LLM Inference framework implements cost-aware layer allocation, achieving Pareto-optimal resource utilisation where no device can improve performance without degrading another's, demonstrated through Lyapunov optimisation, yielding 39% queueing delay reduction under strict deadline constraints [70]. Handling device churn is critical to these approaches: Jin, Du and Chen [62] demonstrates that 31% of edge devices experience intermittent availability, necessitating redundant computation that increases resource consumption by 45% but ensures 99.9% task completion reliability.

Interoperability challenges extend beyond technical compatibility to encompass semantic consistency and quality assurance. The absence of standardised interfaces results in 40-60% integration overhead when combining models from different frameworks [13]. Emerging standards like Open Neural Network Exchange (ONNX) provide format compatibility but fail to address behavioural consistency: identical models exhibit 5-15% performance variation across platforms due to implementation differences in floating-point arithmetic and memory management [147]. The AmoebaLLM framework [35] addresses this through a shape-aware mixture of LoRAs that maintain consistent behaviour across heterogeneous deployments, achieving state-of-the-art accuracy-efficiency trade-offs through one-time fine-tuning that generates platform-specific subnets.

Scalability research must achieve quantitative targets to enable practical deployment. First, communication protocols must reduce overhead to less than 10% of computation time – current implementations consume 30-70% of inference latency on bandwidth-constrained networks [160]. Second, resource allocation algorithms must achieve near-optimal (within 5% of theoretical bounds) utilisation across 1000+ heterogeneous devices – existing approaches scale polynomially, becoming intractable beyond 100 devices [36]. Third, standardisation efforts must establish performance guarantees with less than 1% variation across platforms – critical for safety-critical applications requiring deterministic behaviour [179]. Fourth, adaptive partitioning must respond to resource changes within 100 ms – current reconfiguration requires 1-5 seconds, causing service interruptions [109].

6.5. Emerging applications and future research directions

The convergence of edge computing capabilities and LLM efficiency improvements enables transformative applications across diverse domains, each presenting unique requirements and constraints. Healthcare applications demand sub-second inference for real-time diagnosis while maintaining HIPAA compliance through on-device processing [82]. The integration of LLMs with 6G edge computing, as demonstrated by Zhang et al. [174], enables remote medical consultation with 150 ms latency – sufficient for real-time video analysis – while processing 50,000 patient interactions monthly with $2.3 million annual cost savings through reduced cloud infrastructure. Smart city deployments process 400 billion sensor readings daily across 15 manufacturing facilities, identifying equipment failures 4.2 hours before occurrence with 92.5% precision [136].

Technical integration with existing systems presents multifaceted challenges requiring novel architectural approaches. Zhu et al. [182] proposes multi-agent systems for autonomous coordination, though message passing overhead increases quadratically with agent count, limiting scalability to 50-100 agents. Reinforcement learning integration enables adaptive behaviour: Xu, Niyato and Brinton [146] demonstrates 34% improvement in task completion through policy gradient optimisation, though training requires 10,000 episodes consuming 500 GPU-hours. Graph neural network integration, explored by Fu et al. [34], enables relational reasoning with 87% accuracy on knowledge graph completion tasks, though memory requirements scale $O(n^2)$ with entity count, limiting practical deployment to graphs with less than 1 million nodes.

Multimodal applications represent a frontier where edge deployment offers unique advantages. Wang et al. [133] demonstrates edge-based image captioning, achieving a 94% BLEU-4 score while processing 30 frames per second on mobile devices through selective visual feature extraction. Video summarisation, investigated by Wang et al. [135], reduces 1-hour content to 5-minute summaries with 89% user satisfaction, requiring 8 GB memory for temporal modelling. Cross-lingual applications face additional challenges: Lin et al. [76] shows that edge-based translation systems face significant capacity constraints, achieving 88.4% of cloud model quality through aggressive compression techniques, highlighting the fundamental trade-offs between model performance and deployment feasibility on mobile devices.

Mission-critical applications impose stringent requirements exceeding current edge capabilities. Autonomous driving systems require 10 ms decision latency with 99.999% reliability [91] – current edge LLMs achieve 50 ms latency with 99.9% reliability, necessitating hybrid edge-cloud architectures that violate autonomy requirements. Industrial control systems, analysed by Fakih et al. [31], demand deterministic response times incompatible with probabilistic language models, though recent work on bounded-latency inference shows promise. Healthcare monitoring applications [60] require continuous operation for 168 hours on battery power – current implementations exhaust batteries within 24 hours, highlighting the energy efficiency gap.

Sustainability is a critical consideration given the environmental impact of widespread edge deployment. Strubell, Ganesh and McCallum [122] calculates that training a single LLM produces CO_2 emissions equivalent to 5 cars' lifetime emissions, while edge deployment multiplies this impact across millions of devices. Energy harvesting techniques [65] enable self-powered operation through solar (5-20 mW), vibration (1-10 mW), and thermal (0.1-1 mW) sources, though power budgets remain 10-100× below inference requirements. Workload consolidation [89] reduces energy consumption by 35% through temporal clustering, while renewable energy integration [52] achieves carbon neutrality for 67% of inference workloads through predictive scheduling aligned with renewable availability.

Synthesis of emerging trends reveals five transformative research directions requiring interdisciplinary collaboration. First, neuromorphic computing integration promises 100× energy efficiency improvement through event-driven processing, though requiring fundamental reimagining of transformer architectures [46]. Second, quantum-classical hybrid models could achieve exponential speedup for specific operations, though current quantum devices support only 100-1000 qubits, insufficient for practical LLMs [167]. Third, biological computing using DNA storage and molecular computation offers million-fold density improvements, though with millisecond-scale latency unsuitable for real-time inference [148]. Fourth, swarm intelligence enables collective computation across thousands of simple devices, achieving emergent capabilities exceeding individual components [81]. Fifth, cognitive architectures integrating symbolic reasoning with neural processing promise explainable AI suitable for regulatory compliance, though current implementations increase inference latency by 5-10× [28].

The challenges and opportunities examined in this section collectively define the trajectory of edge-based LLM deployment over the next decade. Current technological capabilities – achieving 40× compression with 5% accuracy loss, supporting 1000-device distributed inference with 450 ms latency, and enabling domain adaptation with 0.1% parameter updates – represent remarkable progress from the monolithic cloud deployments of five years ago. However, the gap between current capabilities and application requirements – demanding 1 ms latency, 99.999% reliability, and indefinite battery operation – necessitates fundamental breakthroughs rather than incremental improvements. The following section synthesises these insights into a coherent vision for the future of edge-based large language models, examining how the convergence of technological advances, application demands, and societal needs will shape this transformative field.

7. Conclusion

This survey has presented a comprehensive analysis of techniques enabling large language model deployment on resource-constrained edge devices, addressing the fundamental mismatch between LLMs' computational demands and edge hardware limitations. Through systematic evaluation of model compression, knowledge distillation, edge-cloud collaboration, and hardware optimisation strategies, we demonstrated that coordinated application of these techniques enables practical deployment of billion-parameter models on devices with limited memory and processing capabilities. The proposed framework bridges the gap between cloud-scale intelligence and edge computing constraints while preserving edge deployment's privacy, latency, and reliability advantages.

Experimental evaluation across multiple architectures and workloads revealed that hybrid edge-microservices implementations achieve 46% reduction in P99 latency and 67% improvement in throughput compared to monolithic deployments, while supporting 10,000 concurrent users within 100 ms response constraints. Knowledge distillation techniques demonstrated 4000× parameter reduction from teacher to student models with less than 3% accuracy degradation on benchmark tasks. Quantisation at 4-bit precision emerged as the optimal trade-off, maintaining 95% of baseline performance while reducing memory footprint by 75% and inference time by 60%. Edge-cloud collaborative frameworks reduced bandwidth consumption by 99.5% through intelligent workload partitioning based on query complexity and network conditions.

These findings fundamentally reshape the deployment landscape for artificial intelligence, enabling privacy-preserving, real-time intelligence at unprecedented scale across healthcare monitoring, autonomous vehicles, and industrial IoT applications. The demonstrated feasibility of edge LLM deployment eliminates the dependency on continuous cloud connectivity, reducing operational costs by 40% while ensuring data sovereignty and regulatory compliance. Multimodal edge intelligence architectures processing vision, audio, and tactile inputs with sub-50 ms latency enable new categories of responsive AI applications previously impossible with cloud-based processing. Current limitations include reduced performance on tasks requiring extensive context beyond 2048 tokens and challenges in supporting languages with limited training representation.

Future research should prioritise four critical directions to advance edge LLM capabilities. Development of neuromorphic computing architectures specifically designed for transformer operations could reduce energy consumption by two orders of magnitude while maintaining current performance levels. Integration of federated learning protocols with homomorphic encryption will enable collaborative model improvement across edge devices without compromising privacy. The investigation of dynamic neural architecture search techniques can produce automatically optimised models for specific edge hardware configurations. Exploration of quantum-inspired compression algorithms promises to reduce model sizes further while potentially improving inference speed through quantum parallelism principles.

Declaration on generative AI: The authors acknowledge the use of artificial intelligence tools during the preparation of this manuscript. Specifically, Scopus AI was employed for initial literature discovery and bibliometric analysis to identify relevant publications in edge computing and LLM domains. Claude Opus 3/4/4.1 assisted in drafting technical sections, organising content structure, and ensuring consistency in terminology throughout the manuscript. Grammarly was used for grammar checking, spell verification, and ensuring adherence to academic English conventions. All AI-generated content underwent rigorous human review, verification, and substantial revision. The authors manually validated all technical claims, verified cited references, confirmed experimental results, and ensured the accuracy of all presented information. The intellectual contributions, critical analysis, synthesis of ideas, and research conclusions are entirely the work of the human authors. The authors take full responsibility for this publication's content, accuracy, and integrity, including any errors or omissions that may exist despite using these assistive tools.

References

[1] Agrawal, R., Kumar, H. and Lnu, S.R., 2025. Efficient LLMs for Edge Devices: Pruning, Quantization, and Distillation Techniques. *2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS)*. pp.1413–1418. Available from: https://doi.org/10.1109/ICMLAS64557.2025.10968787.

[2] Ali, M., Aliagha, E., Elnashar, M. and Göhringer, D., 2025. P-CORE: Exploring RISC-V Packed-SIMD Extension for CNNs. *IEEE Access*, 13, pp.146603–146616. Available from: https://doi.org/10.1109/ACCESS.2025.3600360.

[3] An, Y., Zhao, X., Yu, T., Tang, M. and Wang, J., 2024. Fluctuation-Based Adaptive Structured Pruning for Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10), pp.10865–10873. Available from: https://doi.org/10.1609/aaai.v38i10.28960.

[4] Armeniakos, G., Maras, A., Xydis, S. and Soudris, D., 2025. Mixed-precision Neural Networks on RISC-V Cores: ISA extensions for Multi-Pumped Soft SIMD Operations. *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. New York, NY, USA: Association for Computing Machinery, p.235. Available from: https://doi.org/10.1145/3676536.3676840.

[5] Arriola, M., Gokaslan, A.K., Chiu, J.T., Yang, Z., Qi, Z., Han, J., Sahoo, S.S. and Kuleshov, V., 2025. Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models. *13th International Conference on Learning Representations, ICLR 2025*. pp.84192–84219. Available from: https://openreview.net/forum?id=tyEyYT267x.

[6] Bai, J., Chen, D., Qian, B., Yao, L. and Li, Y., 2025. Federated fine-tuning of large language models under heterogeneous tasks and client resources. *Proceedings*

of the 38th International Conference on Neural Information Processing Systems, NIPS '24. Red Hook, NY, USA: Curran Associates Inc., p.461.

[7] Bandamudi, L., Singh, R.K., Kunde, S., Mishra, M. and Singhal, R., 2024. LLaMPS: Large Language Models Placement System. *Companion of the 15th ACM/SPEC International Conference on Performance Engineering*, ICPE '24 Companion. New York, NY, USA: Association for Computing Machinery, p.87–88. Available from: https://doi.org/10.1145/3629527.3651404.

[8] Basit, A. and Shafique, M., 2024. TinyDigiClones: A Multi-Modal LLM-Based Framework for Edge-optimized Personalized Avatars. *Proceedings of the International Joint Conference on Neural Networks*. Institute of Electrical and Electronics Engineers Inc., pp.1–9. Available from: https://doi.org/10.1109/IJCNN60899.2024.10649909.

[9] Bhardwaj, S., Singh, P. and Pandit, M.K., 2024. A Survey on the Integration and Optimization of Large Language Models in Edge Computing Environments. *2024 16th International Conference on Computer and Automation Engineering, ICCAE 2024*. Institute of Electrical and Electronics Engineers Inc., pp.168–172. Available from: https://doi.org/10.1109/ICCAE59995.2024.10569285.

[10] Bin Son, S., Kim, J., Cho, C. and Park, S., 2025. Trends in Network Optimization Using Large Language Models. *Journal of Korean Institute of Communications and Information Sciences*, 50(7), pp.1073–1084. Available from: https://doi.org/10.7840/kics.2025.50.7.1073.

[11] Bodenham, M. and Kung, J., 2024. Skipformer: Evolving beyond Blocks for Extensively Searching On-Device Language Models with Learnable Attention Window. *IEEE Access*, 12, pp.124428–124439. Available from: https://doi.org/10.1109/ACCESS.2024.3420232.

[12] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D., 2020. Language Models are Few-Shot Learners. In: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin, eds. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. pp.1877–1901. Available from: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[13] Cai, F., Yuan, D., Yang, Z. and Cui, L., 2024. Edge-LLM: A Collaborative Framework for Large Language Model Serving in Edge Computing. In: R.N. Chang, C.K. Chang, Z. Jiang, J. Yang, Z. Jin, M. Sheng, J. Fan, K.K. Fletcher, Q. He, Q. He, C. Ardagna, J. Yang, J. Yin, Z. Wang, A. Beheshti, S. Russo, N. Atukorala, J. Wu, P.S. Yu, H. Ludwig, S. Reiff-Marganiec, E. Zhang, A. Sailer, N. Bena, K. Li, Y. Watanabe, T. Zhao, S. Wang, Z. Tu, Y. Wang and K. Wei, eds. *Proceedings of the IEEE International Conference on Web Services, ICWS*. Institute of Electrical and Electronics Engineers Inc., pp.799–809. Available from: https://doi.org/10.1109/ICWS62655.2024.00099.

[14] Candel, A., McKinney, J., Singer, P., Pfeiffer, P., Jeblick, M., Lee, C.M. and Conde, M.V., 2023. H2O Open Ecosystem for State-of-the-art Large Language Models. In: Y. Feng and E. Lefever, eds. *EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings of the System Demonstrations*. Association for Computational Linguistics (ACL), pp.82–89.

[15] Cao, D. and Aref, S., 2026. Enhancing Ultra-Low-Bit Quantization of Large Language Models Through Saliency-Aware Partial Retraining. In: V. Torra,

Y. Narukawa and J. Domingo-Ferrer, eds. *Modeling Decisions for Artificial Intelligence*, *Lecture notes in computer science*, vol. 15957. Cham: Springer Nature Switzerland, pp.354–383. Available from: https://doi.org/10.1007/978-3-032-00891-6_28.

[16] Chen, H., Zhang, J., Du, Y., Xiang, S., Yue, Z., Zhang, N., Cai, Y. and Zhang, Z., 2024. Understanding the Potential of FPGA-based Spatial Acceleration for Large Language Model Inference. *ACM Transactions on Reconfigurable Technology and Systems*, 18(1), p.5. Available from: https://doi.org/10.1145/3656177.

[17] Chen, K., Zhou, X., Lin, Y., Feng, S., Shen, L. and Wu, P., 2025. A survey on privacy risks and protection in large language models. *Journal of King Saud University - Computer and Information Sciences*, 37(7), p.163. Available from: https://doi.org/10.1007/s44443-025-00177-1.

[18] Chen, Y., Han, Y. and Li, X., 2025. FASTNav: Fine-Tuned Adaptive Small-Language- Models Trained for Multi-Point Robot Navigation. *IEEE Robotics and Automation Letters*, 10(1), pp.390–397. Available from: https://doi.org/10.1109/LRA.2024.3506280.

[19] Chen, Y., Li, R., Yu, X., Zhao, Z. and Zhang, H., 2025. Adaptive layer splitting for wireless large language model inference in edge computing: a model-based reinforcement learning approach. *Frontiers of Information Technology and Electronic Engineering*, 26(2), pp.278–292. Available from: https://doi.org/10.1631/FITEE.2400468.

[20] Chen, Y., Li, R., Zhao, Z., Peng, C., Wu, J., Hossain, E. and Zhang, H., 2024. NetGPT: An AI-Native Network Architecture for Provisioning Beyond Personalized Generative Services. *IEEE Network*, 38(6), pp.404–413. Available from: https://doi.org/10.1109/MNET.2024.3376419.

[21] Deschenaux, J. and Gulcehre, C., 2025. Beyond Autoregression: Fast LLMs via Self-Distillation Through Time. *13th International Conference on Learning Representations, ICLR 2025*. pp.87644–87682. Available from: https://openreview.net/forum?id=uZ5K4HeNwd.

[22] Dettmers, T., Svirschevski, R., Egiazarian, V., Kuznedelev, D., Frantar, E., Ashkboos, S., Borzunov, A., Hoefler, T. and Alistarh, D., 2024. SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression. *12th International Conference on Learning Representations, ICLR 2024*. Available from: https://proceedings.iclr.cc/paper_files/paper/2024/file/1787533e171dcc8549cc2eb5a4840eec-Paper-Conference.pdf.

[23] Devlin, J., Chang, M., Lee, K. and Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: J. Burstein, C. Doran and T. Solorio, eds. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, pp.4171–4186. Available from: https://doi.org/10.18653/V1/N19-1423.

[24] Dhar, N., Deng, B., Islam, M.R., Ahmad Nasif, K.F., Zhao, L. and Suo, K., 2024. Activation Sparsity Opportunities for Compressing General Large Language Models. *2024 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. Available from: https://doi.org/10.1109/IPCCC59868.2024.10850382.

[25] Do, D.T., Shirai, K. and Nguyen, L.M., 2026. WIP: Iterative Post-training Pruning with Weighted Importance Estimation for Large Language Models. In: R. Ichise, ed. *Natural Language Processing and Information Systems*, *Lecture notes in computer science*, vol. 15836. Cham: Springer Nature Switzerland, pp.186–200. Available from: https://doi.org/10.1007/978-3-031-97141-9_13.

[26] Du, C., Wen, Q., Wei, Z. and Zhang, H., 2024. Energy efficient spike transformer accelerator at the edge. *Intelligent Marine Technology and Systems*, 2(1), p.24. Available from: https://doi.org/10.1007/s44295-024-00040-5.

[27] Du, J., Lin, T., Jiang, C., Yang, Q., Bader, C.F. and Han, Z., 2024. Distributed Foundation Models for Multi-Modal Learning in 6G Wireless Networks. *IEEE Wireless Communications*, 31(3), pp.20–30. Available from: https://doi.org/10.1109/MWC.009.2300501.

[28] Dubey, P. and Kumar, M., 2025. Integrating Explainable AI with Federated Learning for Next-Generation IoT: A comprehensive review and prospective insights. *Computer Science Review*, 56, p.100697. Available from: https://doi.org/10.1016/J.COSREV.2024.100697.

[29] Eccles, B.J., Wong, L. and Varghese, B., 2026. Mosaic: Composite projection pruning for resource-efficient LLMs. *Future Generation Computer Systems*, 175, p.108056. Available from: https://doi.org/10.1016/j.future.2025.108056.

[30] Erdogan, L.E., Lee, N., Jha, S., Kim, S., Tabrizi, R., Moon, S., Hooper, C.R.C., Anumanchipalli, G., Keutzer, K. and Gholami, A., 2024. TinyAgent: Function Calling at the Edge. In: D.I. Hernandez Farias, T. Hope and M. Li, eds. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Miami, Florida, USA: Association for Computational Linguistics, pp.80–88. Available from: https://doi.org/10.18653/v1/2024.emnlp-demo.9.

[31] Fakih, M., Dharmaji, R., Moghaddas, Y., Quiros, G., Ogundare, O. and Al Faruque, M.A., 2024. LLM4PLC: Harnessing Large Language Models for Verifiable Programming of PLCs in Industrial Control Systems. *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice*, ICSE-SEIP '24. New York, NY, USA: Association for Computing Machinery, p.192–203. Available from: https://doi.org/10.1145/3639477.3639743.

[32] Flemings, J., Razaviyayn, M. and Annavaram, M., 2024. Differentially Private Next-Token Prediction of Large Language Models. In: K. Duh, H. Gomez and S. Bethard, eds. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Mexico City, Mexico: Association for Computational Linguistics, pp.4390–4404. Available from: https://doi.org/10.18653/v1/2024.naacl-long.247.

[33] Friha, O., Amine Ferrag, M., Kantarci, B., Cakmak, B., Ozgun, A. and Ghoualmi-Zine, N., 2024. LLM-Based Edge Intelligence: A Comprehensive Survey on Architectures, Applications, Security and Trustworthiness. *IEEE Open Journal of the Communications Society*, 5, pp.5799–5856. Available from: https://doi.org/10.1109/OJCOMS.2024.3456549.

[34] Fu, C., Su, Y., Su, K., Liu, Y., Shi, J., Wu, B., Liu, C., Ishi, C.T. and Ishiguro, H., 2025. HAM-GNN: A hierarchical attention-based multi-dimensional edge graph neural network for dialogue act classification. *Expert Syst. Appl.*, 261, p.125459. Available from: https://doi.org/10.1016/J.ESWA.2024.125459.

[35] Fu, Y., Yu, Z., Li, J., Qian, J., Zhang, Y., Yuan, X., Shi, D., Yakunin, R. and Lin, Y.C., 2024. AmoebaLLM: Constructing Any-Shape Large Language Models for Efficient and Instant Deployment. In: A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J.M. Tomczak and C. Zhang, eds. *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*. Available from: http://papers.nips.cc/paper_files/paper/2024/hash/8f11e548311c7fd3f33596a4d1dd41f0-Abstract-Conference.html.

[36] Gao, Z., Zhang, Z., Guo, Y. and Gong, Y., 2025. Federated Adaptive Fine-

Tuning of Large Language Models with Heterogeneous Quantization and LoRA. *Proceedings - IEEE INFOCOM*. Institute of Electrical and Electronics Engineers Inc. Available from: https://doi.org/10.1109/INFOCOM55648.2025.11044641.

[37] Geens, R., Shi, M., Symons, A., Fang, C. and Verhelst, M., 2024. Energy Cost Modelling for Optimizing Large Language Model Inference on Hardware Accelerators. In: G. D., G. U., H. T. and H. K., eds. *International System on Chip Conference*. IEEE Computer Society. Available from: https://doi.org/10.1109/SOCC62300.2024.10737844.

[38] Glint, T., Mittal, B., Sharma, S., Ronak, A.Q., Goud, A., Kasture, N., Momin, Z., Krishna, A. and Mekie, J., 2025. AxLaM: Energy-efficient accelerator design for language models for edge computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 383(2288), p.20230395. Available from: https://doi.org/10.1098/rsta.2023.0395.

[39] Gogineni, K., Suvizi, A. and Venkataramani, G., 2025. LLMs on a Budget: System-Level Approaches to Power-Efficient and Scalable Fine-Tuning. *IEEE Open Journal of the Computer Society*, 6, pp.987–1000. Available from: https://doi.org/10.1109/OJCS.2025.3580498.

[40] Gou, F. and Wu, J., 2024. Optimization of edge server group collaboration architecture strategy in IoT smart cities application. *Peer-to-Peer Networking and Applications*, 17(5), pp.3110–3132. Available from: https://doi.org/10.1007/s12083-024-01739-2.

[41] Graziano, M., Colucci Cante, L. and Di Martino, B., 2025. Deploying Large Language Model on Cloud-Edge Architectures: A Case Study for Conversational Historical Characters. In: L. Barolli, ed. *Advanced Information Networking and Applications*, *Lecture notes on data engineering and communications technologies*, vol. 250. Cham: Springer Nature Switzerland, pp.196–205. Available from: https://doi.org/10.1007/978-3-031-87778-0_19.

[42] Guan, Z., Huang, H., Su, Y., Huang, H., Wong, N. and Yu, H., 2024. APTQ: Attention-aware Post-Training Mixed-Precision Quantization for Large Language Models. *Proceedings of the 61st ACM/IEEE Design Automation Conference*, DAC '24. New York, NY, USA: Association for Computing Machinery, p.107. Available from: https://doi.org/10.1145/3649329.3658498.

[43] Guo, Y., Hao, Z., Shao, J., Zhou, J., Liu, X., Tong, X., Zhang, Y., Chen, Y., Peng, W. and Ma, Z., 2025. PT-BitNet: Scaling up the 1-Bit large language model with post-training quantization. *Neural Networks*, 191, p.107855. Available from: https://doi.org/10.1016/j.neunet.2025.107855.

[44] Habibi, S. and Ercetin, O., 2025. Edge-LLM Inference With Cost-Aware Layer Allocation and Adaptive Scheduling. *IEEE Access*, 13, pp.131614–131637. Available from: https://doi.org/10.1109/ACCESS.2025.3592308.

[45] Han, C., Yang, T., Cui, Z. and Sun, X., 2025. A Privacy-Preserving and Trustworthy Inference Framework for LLM-IoT Integration via Hierarchical Federated Collaborative Computing. *IEEE Internet of Things Journal*. Available from: https://doi.org/10.1109/JIOT.2025.3583764.

[46] Hanchuk, D.O. and Semerikov, S.O., 2024. Automating machine learning: A meta-synthesis of MLOps tools, frameworks and architectures. In: S.O. Semerikov and A.M. Striuk, eds. *Proceedings of the 7th Workshop for Young Scientists in Computer Science & Software Engineering (CS&SE@SW 2024), Virtual Event, Kryvyi Rih, Ukraine, December 27, 2024*, *CEUR workshop proceedings*, vol. 3917. CEUR-WS.org, pp.362–414. Available from: https://ceur-ws.org/Vol-3917/paper60.pdf.

[47] Hao, J., Sun, W., Xin, X., Meng, Q., Chen, Z., Ren, P. and Ren, Z., 2024. MEFT: Memory-Efficient Fine-Tuning through Sparse Adapter. In: L.W. Ku,

A. Martins and V. Srikumar, eds. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, pp.2375–2388. Available from: https://doi.org/10.18653/v1/2024.acl-long.129.

[48] Hao, Z., Jiang, H., Jiang, S., Ren, J. and Cao, T., 2024. Hybrid SLM and LLM for Edge-Cloud Collaborative Inference. *Proceedings of the Workshop on Edge and Mobile Foundation Models*, EdgeFM '24. New York, NY, USA: Association for Computing Machinery, p.36–41. Available from: https://doi.org/10.1145/3662006.3662067.

[49] He, J., Wu, S., Wen, W., Xue, C.J. and Li, Q., 2024. CHESS: Optimizing LLM Inference via Channel-Wise Thresholding and Selective Sparsification. In: Y. Al-Onaizan, M. Bansal and Y.N. Chen, eds. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics, pp.18658–18668. Available from: https://doi.org/10.18653/v1/2024.emnlp-main.1038.

[50] Helmy, M., Khial, N., Yaacoub, E. and Mohamed, A., 2024. OLRAMT-DEC: Online Learning-Based Resource Allocation for AI Model Training in a Device-Edge-Cloud Continuum. *2024 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*. Available from: https://doi.org/10.1109/GCAIOT63427.2024.10833575.

[51] Hong, L., Pan, S., Feng, F. and Jiao, C., 2025. Collaborative Communication for Edge LLM Servicing in Adversarial Networks: An MARL-Empowered Stackelberg Game Approach. *IEEE Internet of Things Journal*, 12(20), pp.41309–41317. Available from: https://doi.org/10.1109/JIOT.2025.3583280.

[52] Hossain, M.S., Hao, Y., Hu, L., Liu, J., Wei, G. and Chen, M., 2024. Immersive Multimedia Service Caching in Edge Cloud with Renewable Energy. *ACM Trans. Multim. Comput. Commun. Appl.*, 20(6), pp.173:1–173:23. Available from: https://doi.org/10.1145/3643818.

[53] Hu, C., Huang, H., Xu, L., Chen, X., Wang, C., Xu, J., Chen, S., Feng, H., Wang, S., Bao, Y., Sun, N. and Shan, Y., 2025. ShuffleInfer: Disaggregate LLM Inference for Mixed Downstream Workloads. *ACM Transactions on Architecture and Code Optimization*, 22(2), p.77. Available from: https://doi.org/10.1145/3732941.

[54] Hu, Y., Wang, Y., Liu, R., Shen, Z. and Lipson, H., 2024. Reconfigurable Robot Identification from Motion Data. *IEEE International Conference on Intelligent Robots and Systems*. Institute of Electrical and Electronics Engineers Inc., pp.14133–14140. Available from: https://doi.org/10.1109/IROS58592.2024.10801809.

[55] Huang, H., Meng, T. and Jia, W., 2025. Joint Optimization of Prompt Security and System Performance in Edge-Cloud LLM Systems. *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*. Available from: https://doi.org/10.1109/INFOCOM55648.2025.11044720.

[56] Huang, Y., Song, J., Wang, Z., Zhao, S., Chen, H., Juefei-Xu, F. and Ma, L., 2025. Look Before You Leap: An Exploratory Study of Uncertainty Analysis for Large Language Models. *IEEE Transactions on Software Engineering*, 51(2), pp.413–429. Available from: https://doi.org/10.1109/TSE.2024.3519464.

[57] Ibrahim, M., Wan, Z., Li, H., Panda, P., Krishna, T., Kanerva, P., Chen, Y. and Raychowdhury, A., 2024. Special Session: Neuro-Symbolic Architecture Meets Large Language Models: A Memory-Centric Perspective. *Proceedings - 2024 International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS 2024*. Institute of Electrical and Electronics Engineers Inc., pp.11–20. Available from: https://doi.org/10.1109/CODES-ISSS60120.2024.00012.

[58] Jain, A.M. and Jain, A., 2025. Scaling LLM Inference Architectures: A Perfor-

mance Analysis for Chatbot Applications. *2025 6th International Conference on Artificial Intelligence, Robotics and Control (AIRC)*. pp.8–16. Available from: https://doi.org/10.1109/AIRC64931.2025.11077484.

[59] Jayanth, R., Gupta, N., Kundu, S., Mathaikutty, D.A. and Prasanna, V., 2024. Towards Real-Time LLM Inference on Heterogeneous Edge Platforms. *2024 IEEE 31st International Conference on High Performance Computing, Data and Analytics Workshop (HiPCW)*. pp.197–198. Available from: https://doi.org/10.1109/HiPCW63042.2024.00076.

[60] Ji, S., Zheng, X., Sun, J., Chen, R., Gao, W. and Srivastava, M., 2024. MindGuard: Towards Accessible and Sitgma-free Mental Health First Aid via Edge LLM. *Corr*, abs/2409.10064. 2409.10064, Available from: https://doi.org/10.48550/ARXIV.2409.10064.

[61] JianHao, Z., Lv, C., Wang, X., Wu, M., Liu, W., Li, T., Ling, Z., Zhang, C., Zheng, X. and Huang, X., 2024. Promoting Data and Model Privacy in Federated Learning through Quantized LoRA. In: Y. Al-Onaizan, M. Bansal and Y.N. Chen, eds. *Findings of the Association for Computational Linguistics: EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, pp.10501–10512. Available from: https://doi.org/10.18653/v1/2024.findings-emnlp.615.

[62] Jin, C., Du, T. and Chen, X., 2025. Energy-Efficient Model Decoupling for Personalized Federated Learning on Cloud-Edge Computing Networks. *Transactions on Emerging Telecommunications Technologies*, 36(7), p.e70203. Available from: https://doi.org/10.1002/ett.70203.

[63] Kawaharazuka, K., Obinata, Y., Kanazawa, N., Okada, K. and Inaba, M., 2023. Robotic Applications of Pre-Trained Vision-Language Models to Various Recognition Behaviors. *IEEE-RAS International Conference on Humanoid Robots*. IEEE Computer Society, pp.1–8. Available from: https://doi.org/10.1109/Humanoids57100.2023.10375211.

[64] Khalfi, M.F. and Tabbiche, M.N., 2025. GPThingSim: A IoT Simulator Based GPT Models Over an Edge-Cloud Environments. *International Journal of Networked and Distributed Computing*, 13(1), p.1. Available from: https://doi.org/10.1007/s44227-024-00045-w.

[65] Khoshsirat, A., Perin, G. and Rossi, M., 2024. Decentralized LLM inference over edge networks with energy harvesting. *Corr*, abs/2408.15907. 2408.15907, Available from: https://doi.org/10.48550/ARXIV.2408.15907.

[66] Kim, J., Seo, M. and Nguyen, X.T., 2025. Mixed INT4-INT8 LLM Quantization via Progressive Layerwise Assignment with Dynamic Sensitivity Estimation. *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*. Available from: https://doi.org/10.1109/ISCAS56072.2025.11043378.

[67] Latif, E., Fang, L., Ma, P. and Zhai, X., 2024. Knowledge Distillation of LLMs for Automatic Scoring of Science Assessments. In: A.M. Olney, I.A. Chounta, Z. Liu, O.C. Santos and I.I. Bittencourt, eds. *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, *Communications in computer and information science*, vol. 2151. Cham: Springer Nature Switzerland, pp.166–174. Available from: https://doi.org/10.1007/978-3-031-64312-5_20.

[68] Lee, K.H., Sim Lai, M., Lim, S.W., Shuang Ru Teh, J., Nizam, S., Nee, Y.K., Keong Koay, E. and Lee, M.S., 2024. Methodologies for Selecting Optimal Hardware for Locally Deployed LLMs Using a Performance, Accuracy and Cost (PAC) Approach. *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*. pp.362–369. Available from: https://doi.org/10.1109/FLLM63129.2024.10852499.

[69] Lenjani, M. and Skadron, K., 2022. Supporting Moderate Data Dependency, Po-

sition Dependency, and Divergence in PIM-Based Accelerators. *IEEE Micro*, 42(1), pp.108–115. Available from: https://doi.org/10.1109/MM.2021.3136189.

[70] Li, J., Li, T., Shen, G., Zhao, D., Zhang, Q. and Zeng, Y., 2025. Pushing up to the Limit of Memory Bandwidth and Capacity Utilization for Efficient LLM Decoding on Embedded FPGA. *2025 Design, Automation & Test in Europe Conference (DATE)*. Available from: https://doi.org/10.23919/DATE64628.2025.10993087.

[71] Li, Q., Shen, Z., Qin, Z., Xie, Y., Zhang, X., Du, T., Cheng, S., Wang, X. and Yin, J., 2024. TransLinkGuard: Safeguarding Transformer Models Against Model Stealing in Edge Deployment. *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24. New York, NY, USA: Association for Computing Machinery, p.3479–3488. Available from: https://doi.org/10.1145/3664647.3680786.

[72] Li, Q., Wen, J. and Jin, H., 2024. Governing Open Vocabulary Data Leaks Using an Edge LLM through Programming by Example. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(4), p.179. Available from: https://doi.org/10.1145/3699760.

[73] Li, Y., Gumaste, D., Turkcan, M.K., Ghaderi, J., Zussman, G. and Kostic, Z., 2025. Distributed VLMs: Efficient Vision-Language Processing through Cloud-Edge Collaboration. *2025 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. pp.280–286. Available from: https://doi.org/10.1109/PerComWorkshops65533.2025.00078.

[74] Liao, M., Chen, W., Shen, J., Guo, S. and Wan, H., 2025. HMoRA: Making LLMs More Effective with Hierarchical Mixture of LoRA Experts. *13th International Conference on Learning Representations, ICLR 2025*. pp.56079–56100. Available from: https://openreview.net/forum?id=lTkHiXeuDl.

[75] Lin, M.G., Wang, J.P., Luo, Y.J. and Wu, A.Y.A., 2024. A 28nm 64.5TOPS/W Sparse Transformer Accelerator with Partial Product-based Speculation and Sparsity-Adaptive Computation. *APCCAS and PrimeAsia 2024*. pp.664–668. Available from: https://doi.org/10.1109/APCCAS62602.2024.10808854.

[76] Lin, Y., Wang, X., Zhang, Z., Wang, M., Xiao, T. and Zhu, J., 2023. MobileNMT: Enabling Translation in 15MB and 30ms. In: S. Sitaram, B.B. Klebanov and J.D. Williams, eds. *Proceedings of the The 61st Annual Meeting of the Association for Computational Linguistics: Industry Track, ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics, pp.368–378. Available from: https://doi.org/10.18653/V1/2023.ACL-INDUSTRY.36.

[77] Lin, Y.J., Chen, K.Y. and Kao, H.Y., 2023. LAD: Layer-Wise Adaptive Distillation for BERT Model Compression. *Sensors*, 23(3), p.1483. Available from: https://doi.org/10.3390/s23031483.

[78] Liu, J., Ponnusamy, P., Cai, T., Guo, H., Kim, Y. and Athiwaratkun, B., 2025. Training-Free Activation Sparsity in Large Language Models. *13th International Conference on Learning Representations, ICLR 2025*. pp.28733–28753. Available from: https://openreview.net/forum?id=dGVZwyq5tV.

[79] Liu, Z., Peng, L., Wang, W., Li, K., Zeng, B., Yu, J. and Liu, X., 2025. Accelerating LLM Inference on RISC-V Edge Devices via Vector Extension Optimization. In: D.S. Huang, C. Zhang, Q. Zhang and Y. Pan, eds. *Advanced Intelligent Computing Technology and Applications*, *Lecture notes in computer science*, vol. 15844. Singapore: Springer Nature Singapore, pp.515–526. Available from: https://doi.org/10.1007/978-981-96-9869-1_43.

[80] Lou, S., Ge, S., Yu, J. and Zhang, G., 2025. TinyVision: Distributed Vision-Language Model with Efficiency and Privacy for Edge Deployment. In: D.S. Huang, W. Chen, Y. Pan and H. Chen, eds. *Advanced Intelligent Computing*

Technology and Applications, *Lecture notes in computer science*, vol. 15851. Singapore: Springer Nature Singapore, pp.175–187. Available from: https://doi.org/10.1007/978-981-96-9849-3_15.

[81] Lua, E.K., Crowcroft, J., Pias, M., Sharma, R. and Lim, S., 2005. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials*, 7(2), pp.72–93. Available from: https://doi.org/10.1109/COMST.2005.1610546.

[82] Ma, M., Gong, C., Zeng, L. and Yang, Y., 2025. Multi-Tier Multi-Node Scheduling of LLM for Collaborative AI Computing. *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*. Available from: https://doi.org/10.1109/INFOCOM55648.2025.11044698.

[83] Ma, W., Yang, X., Zeng, S., Liu, T., Shen, L., Wang, H., Li, S., Wang, J., Zhang, Y., Guo, H., Li, J., Zhang, Z., Zhu, Z., Ning, X., Ho, T.Y., Dai, G. and Wang, Y., 2025. FMC-LLM: Enabling FPGAs for Efficient Batched Decoding of 70B+ LLMs with a Memory-Centric Streaming Architecture. *Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA '25. New York, NY, USA: Association for Computing Machinery, p.55. Available from: https://doi.org/10.1145/3706628.3708863.

[84] Ma, Y., Li, H., Zheng, X., Ling, F., Xiao, X., Wang, R., Wen, S., Chao, F. and Ji, R., 2024. AffineQuant: Affine Transformation Quantization for Large Language Models. *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. Available from: https://openreview.net/forum?id=of2rhALq8l.

[85] Mahr, F., Angeli, G., Sindel, T., Schmidt, K. and Franke, J., 2024. A Reference Architecture for Deploying Large Language Model Applications in Industrial Environments. *IEEE International Symposium for Design and Technology of Electronics Packages, SIITME - Conference Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp.19–23. Available from: https://doi.org/10.1109/SIITME63973.2024.10814877.

[86] Markova, O., Muzyka, I.O., Kuznetsov, D., Kumchenko, Y. and Senko, A., 2024. Enhancing IoT and cyber-physical systems in industry 4.0 through on-premise large language models: real-time data processing, predictive maintenance, and autonomous decision-making. In: M.T.M. Emmerich, V. Lytvyn and V. Vysotska, eds. *Proceedings of the Modern Data Science Technologies Workshop (MoDaST-2024), Lviv, Ukraine, May 31 - June 1, 2024*, *CEUR workshop proceedings*, vol. 3723. CEUR-WS.org, pp.182–197. Available from: https://ceur-ws.org/Vol-3723/paper10.pdf.

[87] Marripudugala, M., 2024. Real-Time IoT Data Analytics Using Advanced Large Language Model Techniques. *2024 Global Conference on Communications and Information Technologies (GCCIT)*. Available from: https://doi.org/10.1109/GCCIT63234.2024.10862622.

[88] Mei, Y., Zhuang, Y., Miao, X., Yang, J., Jia, Z. and Vinayak, R., 2025. Helix: Serving Large Language Models over Heterogeneous GPUs and Network via Max-Flow. *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, ASPLOS '25. New York, NY, USA: Association for Computing Machinery, p.586–602. Available from: https://doi.org/10.1145/3669940.3707215.

[89] Mohiuddin, I. and Almogren, A., 2019. Workload aware VM consolidation method in edge/cloud computing for iot applications. *J. Parallel Distributed Comput.*, 123, pp.204–214. Available from: https://doi.org/10.1016/J.JPDC.2018.09.011.

[90] Monteiro, M., Barros, A., Rodrigues, L., Dermeval, D., Bittencourt, I.I., Isotani, S. and Macario, V., 2025. "Small Device, Big Decision:" Comparing Lightweight

LLMs' Computational Performance and Output Quality for AIED Unplugged. In: A.I. Cristea, E. Walker, Y. Lu, O.C. Santos and S. Isotani, eds. *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium, Blue Sky, and WideAIED*, *Communications in computer and information science*, vol. 2592. Cham: Springer Nature Switzerland, pp.160–167. Available from: https://doi.org/10.1007/978-3-031-99267-4_20.

[91] Mukovoz, V., Vakaliuk, T. and Semerikov, S., 2024. Road Sign Recognition Using Convolutional Neural Networks. *Information Technology for Education, Science, and Technics*, *Lecture notes on data engineering and communications technologies*, vol. 222. Cham: Springer Nature Switzerland, pp.172–188. Available from: https://doi.org/10.1007/978-3-031-71804-5_12.

[92] Nie, B., Shao, Y. and Wang, Y., 2024. Know-Adapter: Towards Knowledge-Aware Parameter-Efficient Transfer Learning for Few-shot Named Entity Recognition. In: N. Calzolari, M.Y. Kan, V. Hoste, A. Lenci, S. Sakti and N. Xue, eds. *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Torino, Italia: ELRA and ICCL, pp.9777–9786.

[93] Ormazabal, A., Artetxe, M. and Agirre, E., 2023. CombLM: Adapting Black-Box Language Models through Small Fine-Tuned Models. In: H. Bouamor, J. Pino and K. Bali, eds. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, pp.2961–2974. Available from: https://doi.org/10.18653/v1/2023.emnlp-main.180.

[94] Ouyang, B., Ye, S., Zeng, L., Qian, T., Li, J. and Chen, X., 2024. Pluto and Charon: A Time and Memory Efficient Collaborative Edge AI Framework for Personal LLMs Fine-tuning. *Proceedings of the 53rd International Conference on Parallel Processing*, ICPP '24. New York, NY, USA: Association for Computing Machinery, p.762–771. Available from: https://doi.org/10.1145/3673038.3673043.

[95] Park, Y., Hyun, J., Kim, H. and Lee, J.W., 2025. DecDEC: A Systems Approach to Advancing Low-Bit LLM Quantization. *Proceedings of the 19th USENIX Symposium on Operating Systems Design and Implementation*. pp.803–819. Available from: https://www.usenix.org/system/files/osdi25-park-yeonhong.pdf.

[96] Piccialli, F., Chiaro, D., Qi, P., Bellandi, V. and Damiani, E., 2025. Federated and edge learning for large language models. *Information Fusion*, 117, p.102840. Available from: https://doi.org/10.1016/j.inffus.2024.102840.

[97] Prabhu, K., Radway, R.M., Yu, J., Bartolone, K., Giordano, M., Peddinghaus, F., Urman, Y., Khwa, W.S., Chih, Y.D., Chang, M.F., Mitra, S. and Raina, P., 2025. MINOTAUR: A Posit-Based 0.42–0.50-TOPS/W Edge Transformer Inference and Training Accelerator. *IEEE Journal of Solid-State Circuits*, 60(4), pp.1311–1323. Available from: https://doi.org/10.1109/JSSC.2025.3545731.

[98] Qiao, D., Ao, X., Liu, Y., Chen, X., Song, F., Qin, Z. and Jin, W., 2025. Tri-AFLLM: Resource-Efficient Adaptive Asynchronous Accelerated Federated LLMs. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(5), pp.4198–4211. Available from: https://doi.org/10.1109/TCSVT.2024.3519790.

[99] Qiao, D., Guo, S., Zhao, J., Le, J., Zhou, P., Li, M. and Chen, X., 2025. ASMAFL: Adaptive Staleness-Aware Momentum Asynchronous Federated Learning in Edge Computing. *IEEE Transactions on Mobile Computing*, 24(4), pp.3390–3406. Available from: https://doi.org/10.1109/TMC.2024.3510135.

[100] Qiao, S., Xu, H., Cao, C., Gong, W., Chen, S. and Liu, J., 2025. PrismPrompt: Layering Prompt-Enhanced Cloud-Edge Collaborative Language Model Toward

Healthcare. *IEEE Network*, 39(4), pp.105–111. Available from: https://doi.org/10.1109/MNET.2025.3532857.

[101] Qiao, Y., Yu, Z., Zhao, Z., Chen, S., Sun, M., Guo, L., Wu, Q. and Liu, J., 2024. VL-Mamba: Exploring State Space Models for Multimodal Learning. In: M. Rezagholizadeh, P. Passban, S. Samiee, V. Partovi Nia, Y. Cheng, Y. Deng, Q. Liu and B. Chen, eds. *Proceedings of The 4th NeurIPS Efficient Natural Language and Speech Processing Workshop*, *Proceedings of machine learning research*, vol. 262. PMLR, pp.102–113. Available from: https://proceedings.mlr.press/v262/qiao24a.html.

[102] Qin, R., Xia, J., Jia, Z., Jiang, M., Abbasi, A., Zhou, P., Hu, J. and Shi, Y., 2024. Enabling On-Device Large Language Model Personalization with Self-Supervised Data Selection and Synthesis. *Proceedings of the 61st ACM/IEEE Design Automation Conference*, DAC '24. New York, NY, USA: Association for Computing Machinery. Available from: https://doi.org/10.1145/3649329.3655665.

[103] Qin, R., Yan, Z., Zeng, D., Jia, Z., Liu, D., Liu, J., Abbasi, A., Zheng, Z., Cao, N., Ni, K., Xiong, J. and Shi, Y., 2025. Robust Implementation of Retrieval-Augmented Generation on Edge-based Computing-in-Memory Architectures. *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '24. New York, NY, USA: Association for Computing Machinery, p.50. Available from: https://doi.org/10.1145/3676536.3676674.

[104] Qin, Y., Wang, Y., Zhao, Z., Yang, X., Zhou, Y., Wei, S., Hu, Y. and Yin, S., 2024. MECLA: Memory-Compute-Efficient LLM Accelerator with Scaling Sub-matrix Partition. *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*. pp.1032–1047. Available from: https://doi.org/10.1109/ISCA59077.2024.00079.

[105] Qu, G., Chen, Q., Wei, W., Lin, Z., Chen, X. and Huang, K., 2025. Mobile Edge Intelligence for Large Language Models: A Contemporary Survey. *IEEE Communications Surveys and Tutorials*. Available from: https://doi.org/10.1109/COMST.2025.3527641.

[106] Qu, W., Zhou, Y., Wu, Y., Xiao, T., Yuan, B., Li, Y. and Zhang, J., 2025. Prompt Inversion Attack Against Collaborative Inference of Large Language Models. *2025 IEEE Symposium on Security and Privacy (SP)*. pp.1695–1712. Available from: https://doi.org/10.1109/SP61157.2025.00160.

[107] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1), January, pp.5485–5551.

[108] Ray, P.P. and Pradhan, M.P., 2025. DLUSEdge: Dynamic Load–Unload Scheduling for Localized LLMs on Resource-Constrained Edge. *KI - Kunstliche Intelligenz*. Available from: https://doi.org/10.1007/s13218-025-00895-8.

[109] Ray, P.P. and Pradhan, M.P., 2025. P2PLLMEdge: Peer-to-Peer Framework for Localized Large Language Models using CPU only Resource-Constrained Edge. *EAI Endorsed Transactions on AI and Robotics*, 4. Available from: https://doi.org/10.4108/airo.9292.

[110] Ren, Y., Zhang, H., Yu, F.R., Li, W., Zhao, P. and He, Y., 2024. Industrial Internet of Things with Large Language Models (LLMs): An Intelligence-based Reinforcement Learning Approach. *IEEE Transactions on Mobile Computing*. Available from: https://doi.org/10.1109/TMC.2024.3522130.

[111] Rjoub, G., Elmekki, H., Islam, S., Bentahar, J. and Dssouli, R., 2025. A hybrid swarm intelligence approach for optimizing Multimodal Large Language Models deployment in edge-cloud-based Federated Learning environments. *Computer*

Communications, 237, p.108152. Available from: https://doi.org/10.1016/j.comcom.2025.108152.

[112] Rong, Y., Mao, Y., He, X. and Chen, M., 2025. Large-Scale Traffic Flow Forecast with Lightweight LLM in Edge Intelligence. *IEEE Internet of Things Magazine*, 8(1), pp.12–18. Available from: https://doi.org/10.1109/IOTM.001.2400047.

[113] Ruan, J., Gao, J., Xie, M., Xiang, S., Yu, Z., Liu, T., Fu, Y. and Qu, X., 2024. GIST: Improving Parameter Efficient Fine-Tuning via Knowledge Interaction. *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24. New York, NY, USA: Association for Computing Machinery, p.8835–8844. Available from: https://doi.org/10.1145/3664647.3680843.

[114] Semerikov, S.O., Vakaliuk, T.A., Kanevska, O.B., Moiseienko, M.V., Donchev, I.I. and Kolhatin, A.O., 2025. LLM on the edge: the new frontier. In: T.A. Vakaliuk and S.O. Semerikov, eds. *Proceedings of the 5th Edge Computing Workshop (doors 2025), Zhytomyr, Ukraine, April 4, 2025*, *CEUR workshop proceedings*, vol. 3943. CEUR-WS.org, pp.137–161. Available from: https://ceur-ws.org/Vol-3943/paper28.pdf.

[115] Shen, X., Dong, P., Lu, L., Kong, Z., Li, Z., Lin, M., Wu, C. and Wang, Y., 2024. Agile-Quant: Activation-Guided Quantization for Faster Inference of LLMs on the Edge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17), pp.18944–18951. Available from: https://doi.org/10.1609/aaai.v38i17.29860.

[116] Shen, X., Han, Z., Lu, L., Kong, Z., Dong, P., Li, Z., Xie, Y., Wu, C., Leeser, M., Zhao, P., Lin, X. and Wang, Y., 2024. HotaQ: Hardware Oriented Token Adaptive Quantization for Large Language Models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. Available from: https://doi.org/10.1109/TCAD.2024.3487781.

[117] Shen, Y., Shao, J., Zhang, X., Lin, Z., Pan, H., Li, D., Zhang, J. and Letaief, K.B., 2024. Large Language Models Empowered Autonomous Edge AI for Connected Intelligence. *IEEE Communications Magazine*, 62(10), pp.140–146. Available from: https://doi.org/10.1109/MCOM.001.2300550.

[118] Shin, J., Yang, H. and Yi, Y., 2025. SparseInfer: Training-free Prediction of Activation Sparsity for Fast LLM Inference. *2025 Design, Automation & Test in Europe Conference (DATE)*. pp.1–7. Available from: https://doi.org/10.23919/DATE64628.2025.10992997.

[119] Sikorski, P., Schrader, L., Yu, K., Billadeau, L., Meenakshi, J., Mutharasan, N., Esposito, F., Aliakbarpour, H. and Babaias, M., 2025. Deployment of Large Language Models to Control Mobile Robots at the Edge. *2025 3rd International Conference on Mechatronics, Control and Robotics (ICMCR)*. pp.19–24. Available from: https://doi.org/10.1109/ICMCR64890.2025.10963303.

[120] Simpson, S.V. and Nagarajan, G., 2021. An edge based trustworthy environment establishment for internet of things: an approach for smart cities. *Wireless Networks*. Available from: https://doi.org/10.1007/s11276-021-02667-2.

[121] Singh, N. and Adhikari, M., 2025. A Hybrid Semi-Asynchronous Federated Learning and Split Learning Strategy in Edge Networks. *IEEE Transactions on Network Science and Engineering*, 12(2), pp.1429–1439. Available from: https://doi.org/10.1109/TNSE.2025.3530999.

[122] Strubell, E., Ganesh, A. and McCallum, A., 2020. Energy and Policy Considerations for Modern Deep Learning Research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09), Apr., pp.13693–13696. Available from: https://doi.org/10.1609/aaai.v34i09.7123.

[123] Sun, B., Huang, Z., Zhao, H., Xiao, W., Zhang, X., Li, Y. and Lin, W., 2024. Llumnix: Dynamic Scheduling for Large Language Model Serving. *Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation,*

OSDI 2024. pp.173–191. Available from: https://www.usenix.org/system/files/osdi24-sun-biao.pdf.

[124] Sun, H., Zhuang, Y., Wei, W., Zhang, C. and Dai, B., 2024. BBox-Adapter: Lightweight Adapting for Black-Box Large Language Models. *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. Available from: https://openreview.net/forum?id=jdRIaUu3xY.

[125] Tambe, T., Zhang, J., Hooper, C., Jia, T., Whatmough, P.N., Zuckerman, J., Santos, M.C.D., Loscalzo, E.J., Giri, D., Shepard, K., Carloni, L., Rush, A., Brooks, D. and Wei, G.Y., 2023. 22.9 A 12nm 18.1TFLOPs/W Sparse Transformer Processor with Entropy-Based Early Exit, Mixed-Precision Predication and Fine-Grained Power Management. *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 2023-February. Institute of Electrical and Electronics Engineers Inc., pp.342–344. Available from: https://doi.org/10.1109/ISSCC42615.2023.10067817.

[126] Tan, F., Lee, R., Dudziak, Ł., Hu, S.X., Bhattacharya, S., Hospedales, T., Tzimiropoulos, G. and Martinez, B., 2024. MobileQuant: Mobile-friendly Quantization for On-device Language Models. In: Y. Al-Onaizan, M. Bansal and Y.N. Chen, eds. *Findings of the Association for Computational Linguistics: EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, pp.9761–9771. Available from: https://doi.org/10.18653/v1/2024.findings-emnlp.570.

[127] Tang, X., Guo, C., Choo, K.K.R. and Liu, Y., 2024. An Efficient and Dynamic Privacy-Preserving Federated Learning System for Edge Computing. *IEEE Transactions on Information Forensics and Security*, 19, pp.207–220. Available from: https://doi.org/10.1109/TIFS.2023.3320611.

[128] Tian, A.X., Zhao, Y., Yin, C., Zhu, W., Tian, X. and Ge, Y., 2024. FanLoRA: Fantastic LoRAs and Where to Find Them in Large Language Model Fine-tuning. In: F. Dernoncourt, D. Preoţiuc-Pietro and A. Shimorina, eds. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Miami, Florida, US: Association for Computational Linguistics, pp.515–528. Available from: https://doi.org/10.18653/v1/2024.emnlp-industry.38.

[129] Tian, C., Qin, X., Tam, K., Li, L., Wang, Z., Zhao, Y., Zhang, M. and Xu, C., 2025. CLONE: Customizing LLMs for Efficient Latency-Aware Inference at the Edge. *Proceedings of the 2025 USENIX Annual Technical Conference*. pp.563–585. Available from: https://www.usenix.org/system/files/atc25-tian.pdf.

[130] Tian, Y., Zhang, B., Tu, Z. and Chu, D., 2025. Adapters Selector: Cross-domains and Multi-tasks LoRA Modules Integration Usage Method. In: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B.D. Eugenio and S. Schockaert, eds. *Proceedings of the 31st International Conference on Computational Linguistics*. Abu Dhabi, UAE: Association for Computational Linguistics, pp.593–605. Available from: https://aclanthology.org/2025.coling-main.40/.

[131] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., 2017. Attention is All you Need. In: I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan and R. Garnett, eds. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. pp.5998–6008. Available from: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[132] Wang, F., Jiang, J., Park, C., Kim, S. and Tang, J., 2025. KaSA: Knowledge-Aware Singular-Value Adaptation of Large Language Models. *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore,*

April 24-28, 2025. OpenReview.net. Available from: https://openreview.net/forum?id=OQqNieeivq.

[133] Wang, N., Xie, J., Luo, H., Cheng, Q., Wu, J., Jia, M. and Li, L., 2023. Efficient Image Captioning for Edge Devices. In: B. Williams, Y. Chen and J. Neville, eds. *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*. AAAI Press, pp.2608–2616. Available from: https://doi.org/10.1609/AAAI.V37I2.25359.

[134] Wang, R. and Li, P., 2024. Semantic are Beacons: A Semantic Perspective for Unveiling Parameter-Efficient Fine-Tuning in Knowledge Learning. In: L.W. Ku, A. Martins and V. Srikumar, eds. *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, pp.9523–9537. Available from: https://doi.org/10.18653/v1/2024.findings-acl.567.

[135] Wang, Y., Dong, Y., Guo, S., Yang, Y. and Liao, X., 2020. Latency-Aware Adaptive Video Summarization for Mobile Edge Clouds. *IEEE Trans. Multim.*, 22(5), pp.1193–1207. Available from: https://doi.org/10.1109/TMM.2019.2939753.

[136] Wang, Y., Zhong, G., Duan, Y., Cheng, Y., Yin, M. and Yang, R., 2025. Efficient and privacy-preserving deep inference towards cloud–edge collaborative. *Applied Soft Computing*, 180, p.113381. Available from: https://doi.org/10.1016/j.asoc.2025.113381.

[137] Wang, Z., Yang, J., Qian, X., Xing, S., Jiang, X., Lv, C. and Zhang, S., 2024. MNN-LLM: A Generic Inference Engine for Fast Large Language Model Deployment on Mobile Devices. *Proceedings of the 6th ACM International Conference on Multimedia in Asia Workshops*, MMAsia '24 Workshops. New York, NY, USA: Association for Computing Machinery, p.11. Available from: https://doi.org/10.1145/3700410.3702126.

[138] Wang, Z., Zhou, Y., Shi, Y. and Letaief, K.B., 2024. Federated Low-Rank Adaptation for Large Language Model Fine-Tuning Over Wireless Networks. *GLOBECOM 2024 - 2024 IEEE Global Communications Conference*. pp.3063–3068. Available from: https://doi.org/10.1109/GLOBECOM52923.2024.10901572.

[139] Wei, X., Zhang, Y., Li, Y., Zhang, X., Gong, R., Guo, J. and Liu, X., 2023. Outlier Suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In: H. Bouamor, J. Pino and K. Bali, eds. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, pp.1648–1665. Available from: https://doi.org/10.18653/v1/2023.emnlp-main.102.

[140] Wu, L., Zhao, Y., Wang, C., Liu, T. and Wang, H., 2024. A First Look at LLM-powered Smartphones. *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering Workshops*, ASEW '24. New York, NY, USA: Association for Computing Machinery, p.208–217. Available from: https://doi.org/10.1145/3691621.3694952.

[141] Wu, Z., Zhi, C., Han, J., Deng, S. and Yin, J., 2025. LLMAppHub: A Large Collection of LLM-based Applications for the Research Community. *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, FSE Companion '25. New York, NY, USA: Association for Computing Machinery, p.1254–1255. Available from: https://doi.org/10.1145/3696630.3731439.

[142] Xia, H., Yang, Z., Dong, Q., Wang, P., Li, Y., Ge, T., Liu, T., Li, W. and Sui, Z., 2024. Unlocking Efficiency in Large Language Model Inference: A Comprehensive Survey of Speculative Decoding. In: L.W. Ku, A. Martins and V. Srikumar, eds. *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok,

Thailand: Association for Computational Linguistics, pp.7655–7671. Available from: https://doi.org/10.18653/v1/2024.findings-acl.456.

[143] Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J. and Han, S., 2023. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato and J. Scarlett, eds. *Proceedings of the 40th International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, vol. 202. PMLR, pp.38087–38099. Available from: https://proceedings.mlr.press/v202/xiao23c.html.

[144] Xie, Q., Zhang, H., Wang, M., Wu, W. and Sun, Z., 2024. Privacy-Enhanced Federated Learning Through Homomorphic Encryption With Cloud Federation. *2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*. pp.1088–1095. Available from: https://doi.org/10.1109/ISPA63168.2024.00144.

[145] Xu, C., Hou, X., Liu, J., Li, C., Huang, T., Zhu, X., Niu, M., Sun, L., Tang, P., Xu, T., Cheng, K.T. and Guo, M., 2023. MMBench: Benchmarking End-to-End Multi-modal DNNs and Understanding Their Hardware-Software Implications. *Proceedings - 2023 IEEE International Symposium on Workload Characterization, IISWC 2023*. Institute of Electrical and Electronics Engineers Inc., pp.154–166. Available from: https://doi.org/10.1109/IISWC59245.2023.00014.

[146] Xu, M., Niyato, D. and Brinton, C.G., 2025. Serving Long-Context LLMs at the Mobile Edge: Test-Time Reinforcement Learning-based Model Caching and Inference Offloading. *CoRR*, abs/2501.14205. 2501.14205, Available from: https://doi.org/10.48550/ARXIV.2501.14205.

[147] Yan, X. and Ding, Y., 2025. Are We There Yet? A Measurement Study of Efficiency for LLM Applications on Mobile Devices. *Proceedings of the 2nd International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things*, FMSys. New York, NY, USA: Association for Computing Machinery, p.19–24. Available from: https://doi.org/10.1145/3722565.3727192.

[148] Yang, M., Yang, Y. and Jiang, P., 2024. A design method for edge–cloud collaborative product service system: a dynamic event-state knowledge graph-based approach with real case study. *International Journal of Production Research*, 62(7), pp.2584–2605. Available from: https://doi.org/10.1080/00207543.2023.2219345.

[149] Yang, T., Li, D., Song, Z., Zhao, Y., Liu, F., Wang, Z., He, Z. and Jiang, L., 2022. DTQAtten: Leveraging Dynamic Token-based Quantization for Efficient Attention Architecture. In: C. Bolchini, I. Verbauwhede and I. Vatajelu, eds. *Proceedings of the 2022 Design, Automation and Test in Europe Conference and Exhibition, DATE 2022*. Institute of Electrical and Electronics Engineers Inc., pp.700–705. Available from: https://doi.org/10.23919/DATE54114.2022.9774692.

[150] Yang, T., Ma, F., Li, X., Liu, F., Zhao, Y., He, Z. and Jiang, L., 2023. DTATrans: Leveraging Dynamic Token-Based Quantization With Accuracy Compensation Mechanism for Efficient Transformer Architecture. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(2), pp.509–520. Available from: https://doi.org/10.1109/TCAD.2022.3181541.

[151] Yang, Y., Huang, X. and Sang, J., 2025. Exploring the Privacy Protection Capabilities of Chinese Large Language Models. *IEEE Multimedia*, 32(2), pp.9–21. Available from: https://doi.org/10.1109/MMUL.2025.3542508.

[152] Yang, Y., Muhtar, D., Shen, Y., Zhan, Y., Liu, J., Wang, Y., Sun, H., Deng, W., Sun, F., Zhang, Q., Chen, W. and Tong, Y., 2025. MTL-LoRA: Low-Rank Adaptation for Multi-Task Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(20), pp.22010–22018. Available from: https://doi.org/10.1609/aaai.v39i20.35509.

[153] Yang, Y., Zhou, J., Wong, N. and Zhang, Z., 2024. LoRETTA: Low-Rank Economic Tensor-Train Adaptation for Ultra-Low-Parameter Fine-Tuning of Large Language Models. In: K. Duh, H. Gomez and S. Bethard, eds. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Mexico City, Mexico: Association for Computational Linguistics, pp.3161–3176. Available from: https://doi.org/10.18653/v1/2024.naacl-long.174.

[154] Yao, K., Gao, P., Li, L., Zhao, Y., Wang, X., Wang, W. and Zhu, J., 2024. Layer-wise Importance Matters: Less Memory for Better Performance in Parameter-efficient Fine-tuning of Large Language Models. In: Y. Al-Onaizan, M. Bansal and Y.N. Chen, eds. *Findings of the Association for Computational Linguistics: EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, pp.1977–1992. Available from: https://doi.org/10.18653/v1/2024.findings-emnlp.109.

[155] Yao, K., Tan, Z., Ye, T., Li, L., Zhao, Y., Liu, W., Wang, W. and Zhu, J., 2025. ScaleOT: Privacy-utility-scalable Offsite-tuning with Dynamic LayerReplace and Selective Rank Compression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(21), pp.22074–22082. Available from: https://doi.org/10.1609/aaai.v39i21.34360.

[156] Yao, Y., Jin, H., Shah, A.D., Han, S., Hu, Z., Stripelis, D., Ran, Y., Xu, Z., Avestimehr, S. and He, C., 2024. ScaleLLM: A Resource-Frugal LLM Serving Framework by Optimizing End-to-End Efficiency. In: F. Dernoncourt, D. Preoţiuc-Pietro and A. Shimorina, eds. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Miami, Florida, US: Association for Computational Linguistics, pp.279–289. Available from: https://doi.org/10.18653/v1/2024.emnlp-industry.22.

[157] Yao, Y., Li, Z. and Zhao, H., 2024. GKT: A Novel Guidance-Based Knowledge Transfer Framework For Efficient Cloud-edge Collaboration LLM Deployment. In: L.W. Ku, A. Martins and V. Srikumar, eds. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL), pp.3433–3446.

[158] Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Chen, C., Li, H., Zhao, W., He, Z., Chen, Q., Zhou, R., Zou, Z., Zhang, H., Hu, S., Zheng, Z., Zhou, J., Cai, J., Han, X., Zeng, G., Li, D., Liu, Z. and Sun, M., 2025. Efficient GPT-4V level multimodal large language model for deployment on edge devices. *Nature Communications*, 16(1), p.5509. Available from: https://doi.org/10.1038/s41467-025-61040-5.

[159] Yao, Z., Tang, Z., Lou, J., Shen, P. and Jia, W., 2024. VELO: A Vector Database-Assisted Cloud-Edge Collaborative LLM QoS Optimization Framework. In: R.N. Chang, C.K. Chang, Z. Jiang, J. Yang, Z. Jin, M. Sheng, J. Fan, K.K. Fletcher, Q. He, Q. He, C. Ardagna, J. Yang, J. Yin, Z. Wang, A. Beheshti, S. Russo, N. Atukorala, J. Wu, P.S. Yu, H. Ludwig, S. Reiff-Marganiec, E. Zhang, A. Sailer, N. Bena, K. Li, Y. Watanabe, T. Zhao, S. Wang, Z. Tu, Y. Wang and K. Wei, eds. *Proceedings of the IEEE International Conference on Web Services, ICWS*. Institute of Electrical and Electronics Engineers Inc., pp.865–876. Available from: https://doi.org/10.1109/ICWS62655.2024.00105.

[160] Ye, S., Ouyang, B., Zeng, L., Qian, T., Chu, X., Tang, J. and Chen, X., 2025. Jupiter: Fast and Resource-Efficient Collaborative Inference of Generative LLMs on Edge Devices. *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*. Available from: https://doi.org/10.1109/INFOCOM55648.2025.11044734.

[161] Yi, H., Lin, F., Li, H., Peiyang, N., Yu, X. and Xiao, R., 2024. Generation Meets Verification: Accelerating Large Language Model Inference with Smart

Parallel Auto-Correct Decoding. In: L.W. Ku, A. Martins and V. Srikumar, eds. *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, pp.5285–5299. Available from: https://doi.org/10.18653/v1/2024.findings-acl.313.

[162] Yokotsuji, R., Lin, D. and Uwano, F., 2024. LLM-Based Interoperable IoT Service Platform. *2024 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. pp.438–444. Available from: https://doi.org/10.1109/WI-IAT62293.2024.00070.

[163] Yu, Z., Liang, S., Ma, T., Cai, Y., Nan, Z., Huang, D., Song, X., Hao, Y., Zhang, J., Zhi, T., Zhao, Y., Du, Z., Hu, X., Guo, Q. and Chen, T., 2024. Cambricon-LLM: A Chiplet-Based Hybrid Architecture for On-Device Inference of 70B LLM. *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*. IEEE Computer Society, pp.1474–1488. Available from: https://doi.org/10.1109/MICRO61859.2024.00108.

[164] Yu, Z., Wang, Z., Li, Y., Gao, R., Zhou, X., Bommu, S.R., Zhao, Y.K. and Lin, Y.C., 2024. EDGE-LLM: Enabling Efficient Large Language Model Adaptation on Edge Devices via Unified Compression and Adaptive Layer Voting. *Proceedings of the 61st ACM/IEEE Design Automation Conference*, DAC '24. New York, NY, USA: Association for Computing Machinery, p.327. Available from: https://doi.org/10.1145/3649329.3658473.

[165] Yu, Z., Wu, S., Jiang, J. and Liu, D., 2024. A knowledge-graph based text summarization scheme for mobile edge computing. *J. Cloud Comput.*, 13(1), p.9. Available from: https://doi.org/10.1186/S13677-023-00585-6.

[166] Yue, Z., Xiang, X., Wang, Y., Guo, R., Han, H., Wei, S., Hu, Y. and Yin, S., 2025. 14.4 A 51.6TFLOPs/W Full-Datapath CIM Macro Approaching Sparsity Bound and <2-30 Loss for Compound AI. *2025 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 68. pp.1–3. Available from: https://doi.org/10.1109/ISSCC49661.2025.10904702.

[167] Zahorodko, P.V., Modlo, Y.O., Kalinichenko, O.O., Selivanova, T.V. and Semerikov, S.O., 2020. Quantum enhanced machine learning: An overview. *Proceedings of the 3rd Workshop for Young Scientists in Computer Science & Software Engineering (CS&SE@SW 2020), Kryvyi Rih, Ukraine, November 27, 2020, CEUR workshop proceedings*, vol. 2832. CEUR-WS.org, pp.94–103. Available from: https://ceur-ws.org/Vol-2832/paper13.pdf.

[168] Zhang, B., Tian, Y., Wang, S., Tu, Z., Chu, D. and Shen, Z., 2024. GongBu: Easily Fine-tuning LLMs for Domain-specific Adaptation. *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24. New York, NY, USA: Association for Computing Machinery, p.5309–5313. Available from: https://doi.org/10.1145/3627673.3679233.

[169] Zhang, D. and Shi, W., 2024. Blockchain-based Edge Intelligence Enabled by AI Large Models for Future Internet of Things. *2024 IEEE 12th International Conference on Information and Communication Networks, ICICN 2024*. Institute of Electrical and Electronics Engineers Inc., pp.368–374. Available from: https://doi.org/10.1109/ICICN62625.2024.10761527.

[170] Zhang, L., Li, B., Thekumparampil, K.K., Oh, S. and He, N., 2024. DPZero: private fine-tuning of language models without backpropagation. *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, p.2446.

[171] Zhang, M., Shen, X., Cao, J., Cui, Z. and Jiang, S., 2024. EdgeShard: Efficient LLM Inference via Collaborative Edge Computing. *IEEE Internet of Things Journal*. Available from: https://doi.org/10.1109/JIOT.2024.3524255.

[172] Zhang, S., 2025. Model collaboration framework design for space-air-ground

integrated networks. *Computer Networks*, 257, p.111013. Available from: https://doi.org/10.1016/j.comnet.2024.111013.

[173] Zhang, S., Ma, Y., Fang, L., Jia, H., D'Alfonso, S. and Kostakos, V., 2024. Enabling On-Device LLMs Personalization with Smartphone Sensing. *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '24. New York, NY, USA: Association for Computing Machinery, p.186–190. Available from: https://doi.org/10.1145/3675094.3677545.

[174] Zhang, T., Xu, X., Wang, Y., Peng, R. and Kadoch, M., 2025. Optimizing Remote Medical Services with AIoT: Integration of Large Language Models and 6G Edge Computing. In: M. Kadoch, M. Cheriet and X. Qiu, eds. *Information Processing and Network Provisioning*, *Communications in computer and information science*, vol. 2416. Singapore: Springer Nature Singapore, pp.278–294. Available from: https://doi.org/10.1007/978-981-96-6468-9_25.

[175] Zhang, Z., Liu, Z., Tian, Y., Khaitan, H., Wang, Z. and Li, S., 2025. R-Sparse: Rank-Aware Activation Sparsity for Efficient LLM Inference. *13th International Conference on Learning Representations, ICLR 2025*. pp.17497–17511. Available from: https://openreview.net/forum?id=VpInEFjoLa.

[176] Zhao, J., Song, Y., Liu, S., Harris, I.G. and Abdu Jyothi, S., 2024. LinguaLinked: Distributed Large Language Model Inference on Mobile Devices. In: Y. Cao, Y. Feng and D. Xiong, eds. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Bangkok, Thailand: Association for Computational Linguistics, pp.160–171. Available from: https://doi.org/10.18653/v1/2024.acl-demos.16.

[177] Zhao, W., Jing, W., Lu, Z. and Wen, X., 2024. Edge and Terminal Cooperation Enabled LLM Deployment Optimization in Wireless Network. *International Conference on Communications in China, ICCC Workshops 2024*. pp.220–225. Available from: https://doi.org/10.1109/ICCCWorkshops62562.2024.10693742.

[178] Zhao, W., Zou, L., Wang, Z., Yao, X. and Yu, B., 2025. HAPE: Hardware-Aware LLM Pruning For Efficient On-Device Inference Optimization. *ACM Transactions on Design Automation of Electronic Systems*, 30(4), July, p.61. Available from: https://doi.org/10.1145/3744244.

[179] Zheng, Y., Chen, Y., Qian, B., Shi, X., Shu, Y. and Chen, J., 2025. A Review on Edge Large Language Models: Design, Execution, and Applications. *ACM Computing Surveys*, 57(8), p.209. Available from: https://doi.org/10.1145/3719664.

[180] Zhong, S., Yang, Z., Gong, R., Wang, R., Huang, R. and Li, M., 2025. ProPD: Dynamic Token Tree Pruning and Generation for LLM Parallel Decoding. *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '24. New York, NY, USA: Association for Computing Machinery, p.202. Available from: https://doi.org/10.1145/3676536.3676695.

[181] Zhou, X., Jia, Q., Hu, Y., Xie, R., Huang, T. and Yu, F.R., 2024. GenG: An LLM-Based Generic Time Series Data Generation Approach for Edge Intelligence via Cross-Domain Collaboration. *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2024*. pp.1–6. Available from: https://doi.org/10.1109/INFOCOMWKSHPS61880.2024.10620716.

[182] Zhu, F., Huang, F., Yu, Y., Liu, G. and Huang, T., 2025. Task Offloading with LLM-Enhanced Multi-Agent Reinforcement Learning in UAV-Assisted Edge Computing. *Sensors*, 25(1), p.175. Available from: https://doi.org/10.3390/s25010175.