LLM on the edge: the new frontier

Serhiy O. Semerikov[1,2,3,4,5], Tetiana A. Vakaliuk[4,3,1,5], Olga B. Kanevska[1], Mykhailo V. Moiseienko[1], Ivan I. Donchev[6] 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*
[6]*South Ukrainian National Pedagogical University named after K. D. Ushynsky, 26 Staroportofrankivska Str., Odesa, 65020, Ukraine*

Abstract

The advent of large language models (LLMs) has revolutionized natural language processing, enabling unprecedented capabilities in text generation, reasoning, and human-machine interaction. However, their deployment on resource-constrained edge devices presents significant challenges due to high computational complexity, large model sizes, and stringent latency and privacy requirements. This survey provides a comprehensive examination of the emerging field of edge-based LLMs, exploring the techniques, frameworks, hardware solutions, and real-world applications that enable their efficient deployment at the edge. We review key strategies such as model quantization, pruning, knowledge distillation, and adapter tuning, alongside edge-cloud collaborative architectures like EdgeShard, Edge-LLM, and PAC. Additionally, we analyze hardware acceleration solutions, including Cambricon-LLM, AxLaM, and DTATrans/DTQAtten, and their role in overcoming resource limitations. The survey highlights diverse applications, from IoT and smart cities to personalized services and multi-modal intelligence, supported by case studies of real-world deployments. Finally, we discuss open challenges – such as resource efficiency, privacy, security, and scalability – and propose future research directions to advance this transformative technology.

Keywords

edge computing, large language models (LLMs), model compression, edge-cloud collaboration, hardware acceleration, IoT applications, personalized services, multi-modal intelligence, privacy-preserving AI, resource efficiency

1. Introduction

The rapid advancements in deep learning and natural language processing have led to the development of large language models (LLMs) that exhibit remarkable performance on a wide range of tasks, from question answering and text generation to reasoning and dialogue [1]. These models, such as GPT-3 [2], BERT [3], and T5 [4], have achieved human-level performance on many benchmarks and have the potential to transform various industries, including healthcare, education, and finance.

However, the deployment of LLMs in real-world scenarios often requires running these models on edge devices, such as smartphones, IoT sensors, and embedded systems [5]. This is motivated by several factors, including the need for low-latency inference in applications like virtual assistants and real-time translation [6], the desire to preserve user privacy by processing data locally, without sending it to

doors-2025: 5th Edge Computing Workshop, April 4, 2025, Zhytomyr, Ukraine
semerikov@gmail.com (S. O. Semerikov); tetianavakaliuk@acnsci.org (T. A. Vakaliuk); o.b.kanevska@gmail.com (O. B. Kanevska); seliverst17moiseenko@gmail.com (M. V. Moiseienko); donchev@pdpu.edu.ua (I. I. Donchev); 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/mvmoiseienko.html (M. V. Moiseienko)
0000-0003-0789-0272 (S. O. Semerikov); 0000-0001-6825-4697 (T. A. Vakaliuk); 0000-0003-1703-7929 (O. B. Kanevska); 0000-0003-4401-0297 (M. V. Moiseienko); 0000-0002-3373-6562 (I. I. Donchev); 0000-0002-3125-3137 (A. O. Kolhatin)

the cloud [7], and the requirement for offline functionality in scenarios with limited or no internet connectivity [8].

In fact, deploying LLMs on edge devices is challenging due to their large model sizes, high computational requirements, and memory footprint [9]. For example, GPT-3 has 175 billion parameters and requires 350GB of memory for inference [10]. This far exceeds the resources available on most edge devices, which typically have limited CPU/GPU capabilities, memory (a few GB), and power constraints.

To address these challenges, there has been a growing interest in developing techniques and frameworks for the efficient deployment of LLMs on edge devices [11, 6, 12]. This survey aims to provide an overview of this emerging field, covering the key aspects of edge-based LLMs, from model compression and acceleration techniques to collaborative frameworks and hardware solutions.

The rest of this survey is organized as follows. Section 2 provides an overview of LLMs for edge deployment, discussing the key characteristics, challenges, and popular open-source frameworks. Section 3 reviews the edge-cloud collaborative frameworks and architectures for optimized LLM inference. Section 4 surveys the hardware acceleration solutions and chipsets for efficient LLM execution on edge devices. Section 5 highlights the real-world applications and systems leveraging edge-based LLMs across various domains. Section 6 discusses the open challenges, opportunities, and future research directions in this field. Finally, section 7 concludes the survey and provides an outlook on the future of edge-based LLMs.

2. Overview of LLMs for edge deployment

Large language models (LLMs) have demonstrated remarkable performance in various natural language processing tasks, such as text generation, question answering, and sentiment analysis [10, 3, 4]. However, deploying these models on edge devices presents unique challenges due to the resource constraints and heterogeneity of edge environments [5, 8].

Deploying LLMs on edge devices presents several significant challenges due to their inherent characteristics. One prominent difficulty stems from their substantial model size. LLMs frequently encompass hundreds of millions, if not billions, of parameters, which translates into considerable storage and memory demands. For instance, GPT-3, with its 175 billion parameters, requires approximately 350GB of memory for inference – a requirement that far exceeds the capacity of most edge devices, typically limited to just a few gigabytes of memory, as noted by Brown et al. [10] and Zhang et al. [9]. Another hurdle arises from the high computational complexity involved in LLM inference. This process relies on intricate matrix operations and attention mechanisms, resulting in substantial computational overhead, as described by Vaswani et al. [13]. Such demands strain the limited CPU and GPU capabilities of edge devices, further compounded by their restrictive power budgets, a challenge highlighted by Shen et al. [11]. Latency requirements also pose a critical concern. Many edge applications, such as virtual assistants and real-time translation tools, depend on rapid inference to ensure a seamless user experience, according to Cai et al. [6]. Yet, the computational intensity of LLMs often introduces significant delays, particularly on resource-constrained hardware, as Yu et al. [12] observes, making it difficult to meet these stringent timing needs. Finally, privacy considerations add another layer of complexity. Edge devices frequently process sensitive user data, and transmitting this information to the cloud for inference can raise substantial privacy risks, as Li et al. [7] points out. Consequently, there is a pressing need for methods that facilitate local inference on edge devices while safeguarding user privacy, an issue Qiao et al. [14] emphasizes as essential for practical deployment.

To address these challenges, various techniques and frameworks have been proposed for the efficient deployment of LLMs on edge devices. These include model compression and acceleration techniques (section 2.2), edge-cloud collaborative frameworks (section 3), and hardware acceleration solutions (section 4).

2.1. Popular open-source LLMs and frameworks

Several open-source LLMs and frameworks have been developed to facilitate the deployment of LLMs on edge devices. These frameworks provide pre-trained models, tools, and techniques for efficient inference and adaptation to edge environments. Table 1 provides an overview of popular open-source LLMs and frameworks for edge deployment.

Table 1
Popular open-source LLMs and frameworks for edge deployment.

Framework	Model	Key features	References
TinyAgent	TinyAgent-1.1B/7B	Accurate function calling, efficient inference	Erdogan et al. [15]
MNN-LLM	MNN-LLM	Quantization, DRAM-Flash hybrid storage	Wang et al. [16]
h2oGPT	h2oGPT (7-70B)	Fine-tuned models, no-code GUI	Candel et al. [17]

TinyAgent [15] is a framework for training and deploying task-specific small language model agents on edge devices. It provides pre-trained models, TinyAgent-1.1B and TinyAgent-7B, which achieve accurate function calling and efficient inference by leveraging techniques like tool retrieval and quantization. TinyAgent also supports real-time interaction through voice input and output, making it suitable for building intelligent personal assistants on edge devices.

MNN-LLM [16] is a generic inference engine for fast LLM deployment on mobile devices. It addresses the memory and latency challenges of edge inference by employing techniques like model quantization and DRAM-Flash hybrid storage. MNN-LLM optimizes the inference process based on the characteristics of mobile CPUs and GPUs, achieving significant speedups compared to other mobile-friendly LLM frameworks.

h2oGPT [17] is an open-source ecosystem for state-of-the-art LLMs, providing a family of fine-tuned models ranging from 7B to 70B parameters. It includes the H2O LLM Studio, a no-code GUI for efficient model fine-tuning, evaluation, and deployment using advanced techniques. h2oGPT models are designed to be scalable and adaptable to various edge environments, making them suitable for a wide range of applications.

Figure 1 illustrates the comparative performance of these frameworks on a benchmark dataset, highlighting their inference speed and memory efficiency on edge devices.

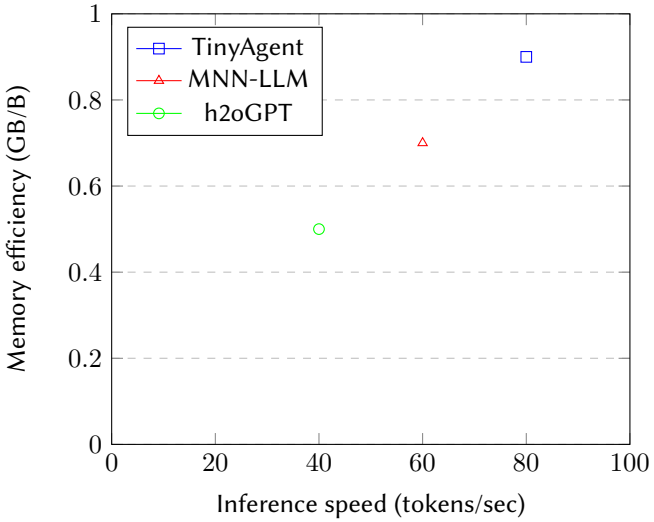

Figure 1: Comparative performance of open-source LLM frameworks on edge devices.

As shown in figure 1, TinyAgent achieves the highest inference speed and memory efficiency among the compared frameworks, making it a promising choice for real-time applications on resource-

constrained devices. MNN-LLM and h2oGPT also demonstrate competitive performance, with MNN-LLM focusing on mobile-specific optimizations and h2oGPT providing a wide range of fine-tuned models for different scenarios.

The selection of an appropriate framework depends on factors such as the target application, device capabilities, and performance requirements.

2.2. Techniques for efficient LLM deployment on edge devices

Various techniques have been proposed for model compression and acceleration to enable the efficient deployment of LLMs on resource-constrained edge devices. These techniques aim to reduce the model size, computational complexity, and memory footprint of LLMs while maintaining their performance and generalization capabilities. Table 2 summarizes the key techniques for efficient LLM deployment on edge devices.

Table 2
Techniques for efficient LLM deployment on edge devices.

Technique	Key idea	References
Quantization	Reducing the precision of model weights and activations	Shen et al. [18], Tambe et al. [19]
Pruning	Removing redundant or less important model parameters	Yu et al. [12], Bhardwaj et al. [8]
Knowledge distillation	Transferring knowledge from a large teacher model to a small student model	Sanh et al. [20]
Adapter tuning	Fine-tuning small adapter modules while keeping the base model fixed	Houlsby et al. [21]

Quantization and pruning are two widely used techniques for reducing the model size and computational complexity of LLMs [18, 19, 12]. Quantization reduces the precision of model weights and activations from 32-bit floating-point to lower bit-widths (e.g., 8-bit or 4-bit integers), resulting in smaller model sizes and faster inference. Pruning removes redundant or less important model parameters, leading to sparse models with reduced computational and memory requirements.

For example, Shen et al. [18] propose HotaQ, a hardware-oriented token adaptive quantization framework for LLMs. HotaQ achieves 4/8-bit quantization for weights and activations while maintaining performance comparable to full-precision models. Yu et al. [12] introduce EDGE-LLM, a layer-wise unified compression technique that generates pruning sparsity and quantization bit-width policies, achieving significant speedups and memory savings on edge devices.

Knowledge distillation [22] is a technique for transferring knowledge from a large teacher model to a small student model, enabling the deployment of compact and efficient models on edge devices. The student model is trained to mimic the behaviour of the teacher model by minimizing the divergence between their output distributions.

Sanh et al. [20] propose DistilBERT, a distilled version of BERT that achieves 95% of the teacher's performance while being 40% smaller and 60% faster. Jiao et al. [23] introduce TinyBERT, a two-stage distillation framework that transfers knowledge from the pre-trained BERT model to a smaller student model, achieving competitive performance on various NLP tasks.

Adapter tuning [21] is a parameter-efficient fine-tuning technique that adapts pre-trained LLMs to downstream tasks by training small adapter modules while keeping the base model fixed. This approach reduces the memory and computational requirements of fine-tuning, making it suitable for edge deployment.

Qiao et al. [14] propose Tri-AFLLM, an adaptive asynchronous accelerated federated LLM framework that leverages adapter tuning for efficient edge deployment. Tri-AFLLM updates only the adapter parameters while keeping the base model frozen, achieving significant resource efficiency and accuracy improvements in federated learning scenarios.

Figure 2 visualizes the impact of these techniques on the model size and inference speed of LLMs, highlighting their effectiveness in enabling edge deployment.

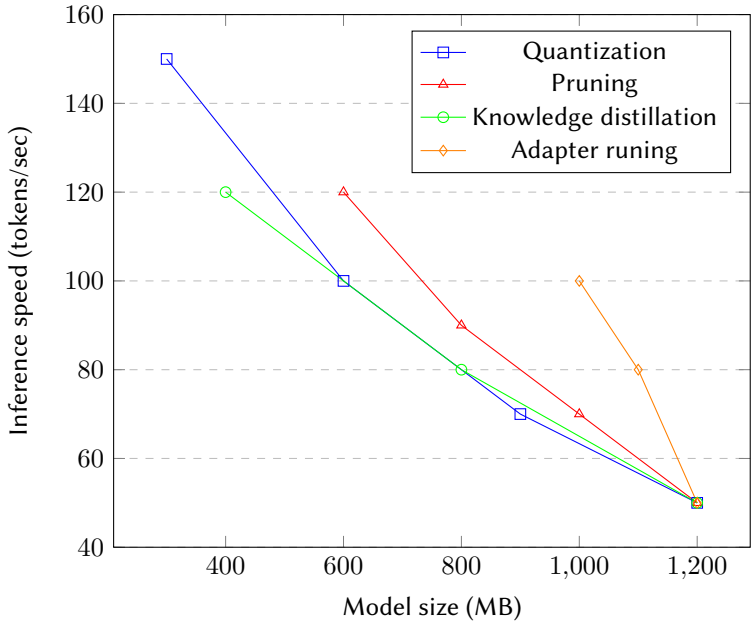

Figure 2: Impact of efficient deployment techniques on LLM model size and inference speed.

As shown in figure 2, all four techniques effectively reduce the model size and improve the inference speed of LLMs, with quantization and knowledge distillation achieving the most significant improvements. Pruning and adapter tuning also demonstrate considerable benefits, especially when combined with other techniques.

The choice of technique depends on various factors, such as the target device's capabilities, performance requirements, and the availability of pre-trained models or training data.

3. Edge-cloud collaborative frameworks and architectures

Deploying large language models on edge devices often necessitates a partnership between edge and cloud environments to address the inherent resource constraints and performance limitations of standalone devices, as explored by Zhou et al. [24] and Zhao et al. [25]. These edge-cloud collaborative frameworks harness the distinct advantages of both edge and cloud computing, paving the way for efficient and scalable LLM inference tailored to real-world applications, a concept further elaborated by Yao et al. [26] and Cai et al. [6].

The drive toward edge-cloud collaboration in LLM deployment stems from several compelling factors. One key incentive is scalability. By distributing computational workloads across multiple edge devices and cloud servers, these frameworks enable LLMs to operate effectively in expansive, large-scale settings, as Zhang et al. [9] demonstrates. Another advantage lies in efficiency. Offloading computationally demanding tasks to the cloud while conducting local inference on edge devices allows these systems to enhance overall performance and optimize resource use, a benefit underscored by Yao et al. [27]. Adaptability also plays a crucial role. This collaboration permits LLM inference to adjust dynamically to fluctuating network conditions, varying device capabilities, and diverse application needs, according to Cai et al. [6]. Privacy emerges as yet another vital consideration. Such frameworks can safeguard sensitive data by processing it locally on edge devices and reserving the cloud for handling non-sensitive computations, a strategy highlighted by Li et al. [7].

Despite these advantages, crafting effective edge-cloud collaborative frameworks for LLM deployment is far from straightforward, presenting a range of obstacles. Heterogeneity stands out as a significant

hurdle. Edge devices vary widely in their hardware and software configurations, complicating the creation of frameworks capable of leveraging these diverse resources efficiently, as Friha et al. [5] points out. Communication overhead poses another challenge. Transferring data and model parameters between edge devices and the cloud can generate substantial delays, particularly when network bandwidth is limited, an issue raised by Zhang et al. [9]. Synchronization adds further complexity. Maintaining consistency and coherence in model updates across edge devices and the cloud is essential to ensure the stability and performance of these collaborative systems, a concern emphasized by Qiao et al. [14]. Security, too, demands attention. The interplay between edge and cloud introduces risks such as data breaches and adversarial attacks during data transfer and processing across platforms, a problem noted by Nazari et al. [28].

To tackle these issues, researchers have put forward a variety of edge-cloud collaborative frameworks and architectures in the literature. These efforts focus on critical aspects of LLM deployment, including model partitioning, task offloading, and resource allocation, as evidenced by the work of Yao et al. [26], Cai et al. [6], and Zhang et al. [9]. Such innovations aim to bridge the gap between the potential of LLMs and the practical realities of edge environments, fostering robust and adaptable solutions.

3.1. Overview of collaborative frameworks

Table 3 provides an overview of representative edge-cloud collaborative frameworks for LLM deployment, highlighting their key features and target applications.

Table 3
Overview of edge-cloud collaborative frameworks for LLM deployment.

Framework	Key features	Target applications	References
EdgeShard	Model partitioning, adaptive device selection	Content generation, decision making	Zhang et al. [9]
Edge-LLM	Adaptive quantization, FM cache, VDF scheduling	AI applications, QoS optimization	Cai et al. [6]
PAC	Parallel adapters, activation cache, data parallelism	Personal LLM fine-tuning	Ouyang et al. [29]

EdgeShard, as introduced by Zhang et al. [9], offers a collaborative edge computing framework designed to facilitate the efficient deployment of computationally demanding large language models on resource-constrained edge devices. This framework achieves its goals by dividing the LLM into smaller, manageable shards, distributing them across edge devices and cloud servers according to their computational capacities and prevailing network conditions. To enhance system performance, EdgeShard incorporates an adaptive algorithm that jointly optimizes device selection and model partitioning, aiming to reduce inference latency while boosting throughput.

- The framework splits the LLM into smaller shards, enabling their execution on edge devices despite limited resources. This partitioning approach carefully considers the computational complexity and memory demands of various model components, such as attention layers and feedforward networks, ensuring efficient operation.
- EdgeShard dynamically identifies the most suitable edge devices and cloud servers for collaborative inference. This selection process adapts to current resource availability, network states, and input data characteristics, ensuring optimal participation across the system.
- To streamline performance, EdgeShard employs a dynamic programming algorithm focused on minimizing end-to-end inference latency and maximizing system throughput. This method accounts for factors like communication overhead, computation duration, and memory limitations inherent to each device.

EdgeShard finds application in diverse LLM-based tasks, including content generation and intelligent

decision-making within IoT systems. Its implementation has yielded notable enhancements in latency and throughput, outperforming traditional cloud-centric deployment strategies.

Edge-LLM, presented by Cai et al. [6], serves as a collaborative framework tailored for large-scale language model deployment within edge computing contexts. By tapping into the computational strengths of both edge devices and cloud servers, it accelerates LLM fine-tuning and inference under resource-limited conditions, all while prioritizing quality of service (QoS) for end users.

- The framework adopts an adaptive quantization strategy, dynamically tailoring the precision of model weights and activations. This adjustment aligns with the computational capabilities of edge devices and the specific needs of the application, striking a balance between inference speed and accuracy.
- Edge-LLM integrates a frequency-based model (FM) cache mechanism, storing frequently accessed model parameters and intermediate results directly on edge devices. This approach cuts down on communication overhead and reduces latency during collaborative inference.
- A value density first (VDF) scheduling algorithm guides Edge-LLM in prioritizing compute-heavy tasks for execution on edge devices with superior capabilities. Less demanding tasks are shifted to the cloud, optimizing resource use and overall system performance.

Edge-LLM has proven effective across various AI applications, including natural language processing and computer vision. Compared to conventional edge computing methods, it delivers marked improvements in computational speed, task throughput, and GPU overhead reduction.

PAC, short for Pluto and Charon, as outlined by Ouyang et al. [29], stands out as a time- and memory-efficient collaborative edge AI framework focused on personal LLM fine-tuning. It harnesses the computational resources of nearby edge devices to enable on-the-spot fine-tuning of personalized LLMs, minimizing communication demands while upholding data privacy.

- PAC introduces parallel adapters, a novel fine-tuning technique that adapts pre-trained LLMs to individual preferences and domains. By training small adapter modules concurrently while keeping the base model unchanged, this method reduces the computational and memory burdens of the fine-tuning process.
- An activation cache mechanism enhances efficiency in PAC by storing intermediate activations from the base model during the forward pass. This storage eliminates redundant computations in the backward pass, streamlining the fine-tuning of parallel adapters.
- The framework blends data parallelism with pipeline parallelism to distribute fine-tuning workloads across proximate edge devices. This hybrid approach minimizes communication overhead and maximizes the use of available computational resources.

PAC excels in personal LLM applications, such as tailored language understanding and generation. Its deployment has demonstrated substantial gains in fine-tuning speed and memory efficiency, surpassing existing state-of-the-art techniques.

Figure 3 provides a comparative analysis of the three collaborative frameworks in terms of their inference latency, throughput, and memory efficiency on representative edge devices and cloud servers.

As shown in figure 3, all three frameworks achieve significant improvements in inference latency, throughput, and memory efficiency compared to traditional cloud-based deployment approaches. EdgeShard demonstrates the lowest latency, while PAC achieves the highest throughput, and Edge-LLM exhibits the best memory efficiency. The choice of framework depends on the specific requirements and constraints of the target application and deployment scenario.

3.2. Techniques for optimized LLM inference

To further optimize the performance and efficiency of LLM inference in edge-cloud collaborative frameworks, various techniques have been proposed in the literature, focusing on aspects such as

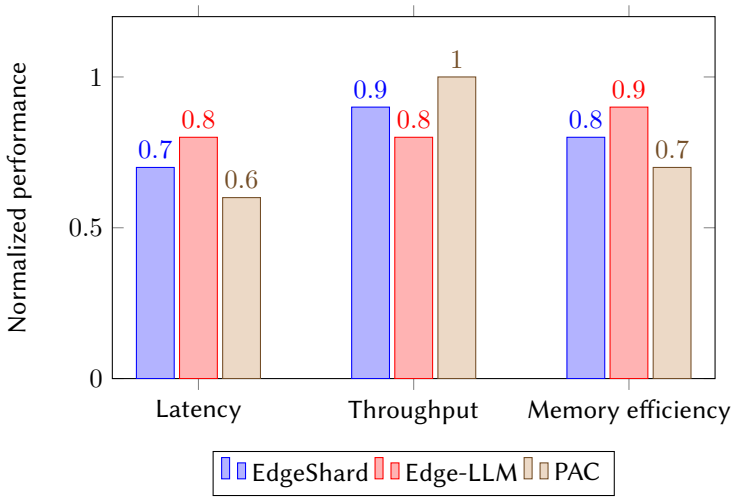

Figure 3: Comparative analysis of collaborative frameworks for LLM deployment on edge devices.

Table 4
Techniques for optimized LLM inference in collaborative frameworks.

Technique	Key idea	References
Adaptive quantization	Dynamically adjusting the precision of model weights and activations	Cai et al. [6], Shen et al. [18]
Reinforcement learning-based scheduling	Learning optimal task scheduling policies through interaction with the environment	Yao et al. [27]
Vector databases and caching	Storing and retrieving frequently accessed model parameters and intermediate results	Yao et al. [27], Cai et al. [6]

adaptive quantization, scheduling, and caching. Table 4 summarizes the key techniques for optimized LLM inference in collaborative frameworks.

Adaptive quantization and scheduling are two key techniques for optimizing the performance and resource utilization of LLM inference in edge-cloud collaborative frameworks. Adaptive quantization dynamically adjusts the precision of model weights and activations based on the computational capabilities of edge devices and the requirements of the target application, achieving a balance between inference speed and accuracy [6, 18]. For example, Cai et al. [6] propose an adaptive quantization strategy in the Edge-LLM framework, which dynamically selects the optimal quantization scheme for each model layer based on the computational capabilities of the target edge device and the performance requirements of the application.

Reinforcement learning-based scheduling learns optimal task scheduling policies through interaction with the environment, considering factors such as the communication overhead, computation time, and resource constraints of each device [27]. Yao et al. [27] introduce a reinforcement learning-based scheduling algorithm in the VELO framework, which learns to optimize the task offloading decisions and resource allocation policies through trial and error, adapting to the dynamic network conditions and workload characteristics.

Vector databases and caching mechanisms are essential for reducing the communication overhead and latency of LLM inference in edge-cloud collaborative frameworks. Vector databases store and retrieve frequently accessed model parameters and intermediate results, enabling efficient reuse of computations across different inference requests [27]. Caching mechanisms, such as the FM cache in Edge-LLM [6], store recently accessed data and models on edge devices, reducing the need for redundant data transfers and computations.

For example, Yao et al. [27] propose a vector database-assisted caching mechanism in the VELO framework, which stores the results of recent LLM inference requests on edge devices and reuses

them for similar requests in the future, significantly reducing the response time and computational cost of the system. Cai et al. [6] introduce the FM cache mechanism in Edge-LLM, which maintains a frequency-based model cache on edge devices, storing the most frequently accessed model parameters and intermediate results for fast retrieval and reuse.

Figure 4 illustrates the impact of these optimization techniques on the inference latency and throughput of LLMs in edge-cloud collaborative frameworks, highlighting their effectiveness in improving the performance and efficiency of the system.

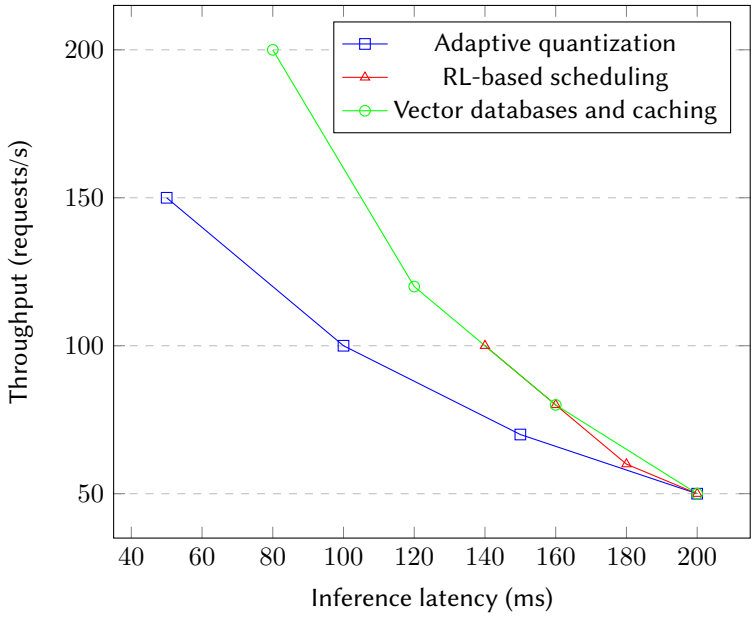

Figure 4: Impact of optimization techniques on LLM inference latency and throughput in collaborative frameworks.

As shown in figure 4, all three optimization techniques effectively reduce the inference latency and improve the throughput of LLMs in edge-cloud collaborative frameworks. Adaptive quantization achieves the most significant latency reduction, while vector databases and caching mechanisms demonstrate the highest throughput improvement. Reinforcement learning-based scheduling also shows considerable benefits in terms of both latency and throughput optimization.

The choice of optimization technique depends on the specific characteristics and requirements of the target application and deployment scenario. In practice, these techniques are often used in combination to achieve the best performance and efficiency for LLM inference in edge-cloud collaborative frameworks.

4. Hardware acceleration solutions and chipsets

Deploying large language models on edge devices demands efficient hardware acceleration solutions to navigate the computational and memory limitations inherent to these resource-constrained platforms, as noted by Bhardwaj et al. [8]. Crafting and implementing such accelerators for edge-based LLMs, however, is no small feat, entailing a complex interplay of challenges and requirements. Achieving high computational performance stands as a primary concern. These accelerators need to support real-time inference and response generation, all while adhering to the stringent power and thermal boundaries of edge devices, a point emphasized by Yu et al. [30]. Equally critical is the pursuit of energy efficiency. To extend battery life and curtail operational costs, LLM accelerators must minimize power consumption for each computational task, a goal highlighted by Glint et al. [31]. Memory efficiency also plays a pivotal role. Given the scarce on-chip and off-chip memory resources available, these accelerators are tasked with optimizing the memory footprint and bandwidth demands of the models, as Wang et al.

[16] underscores. Flexibility emerges as another essential requirement. With the diverse array of LLM architectures and applications in play, edge accelerators must offer programmability, scalability, and adaptability to accommodate varying workload characteristics, according to Tambe et al. [19]. Lastly, minimizing latency is crucial. The accelerators must streamline the entire inference pipeline – from data transfer to preprocessing and postprocessing – to deliver responsive and interactive user experiences; a necessity pointed out by Yang et al. [32]. To meet these multifaceted challenges and requirements, researchers and industry experts have proposed an assortment of hardware acceleration solutions and chipsets. These innovations draw on techniques like specialized processing units, optimized memory hierarchies, and tailored dataflow architectures, as evidenced by the work of Yu et al. [30], Glint et al. [31], and Yang et al. [32], paving the way for more effective LLM deployment at the edge.

4.1. Overview of hardware solutions

Table 5 provides an overview of representative hardware acceleration solutions and chipsets for edge LLM deployment, highlighting their key features and performance characteristics.

Table 5
Overview of hardware acceleration solutions and chipsets for edge LLM deployment.

Solution	Key features	Performance	References
Cambricon-LLM	Hybrid architecture, NPU, NAND flash	3.44 tokens/s (70B), 36.34 tokens/s (7B)	Yu et al. [30]
AxLaM	POSIT-based multipliers, HBM	9x energy reduction, 58% area reduction	Glint et al. [31]
DTATrans/DTQAtten	Dynamic mixed-precision, VSSA	16.04x speedup, 3.62x energy saving	Yang et al. [32]

Cambricon-LLM, as introduced by Yu et al. [30], presents a chiplet-based hybrid architecture tailored for on-device inference of large language models with up to 70 billion parameters. This design integrates a neural processing unit (NPU) with a dedicated NAND flash chip, striking a balance between high performance and energy efficiency while reducing the data movement overhead between processing and memory elements. The framework capitalizes on the NPU's robust computing power alongside the substantial data capacity of the NAND flash, enabling efficient execution of large-scale LLMs directly on edge devices. To further enhance its capability, the NAND flash chip incorporates innovative in-flash computing and on-die error correction code (ECC) techniques. These advancements facilitate lightweight processing within the chip itself, significantly cutting down on data transfer demands between the NPU and flash storage.

Additionally, Cambricon-LLM employs a hardware-tiling strategy to streamline data movement and computation scheduling between these components. This approach minimizes memory access latency and maximizes resource utilization, ensuring optimal performance. In practice, Cambricon-LLM delivers an impressive on-device inference speed of 3.44 tokens per second for 70B LLMs and 36.34 tokens per second for 7B LLMs – performance levels that surpass existing flash-offloading technologies by 22 to 45 times – demonstrating its prowess in enabling large-scale LLM deployment on edge devices.

AxLaM, detailed by Glint et al. [31], emerges as an energy-efficient accelerator crafted for language models on edge devices, harnessing approximate fixed-point POSIT-based multipliers and high bandwidth memory (HBM) to deliver both high performance and low power consumption. This design leverages POSIT-based multipliers to simplify the computational complexity and energy demands of matrix operations central to language models, all while preserving acceptable accuracy levels. Complementing this, AxLaM incorporates high bandwidth memory to efficiently manage the storage and retrieval of model parameters and intermediate activations. This setup reduces memory access latency, thereby boosting overall performance. The accelerator also features a dataflow architecture meticulously tuned to the specific needs of language model workloads. By optimizing the flow of data, AxLaM maximizes the efficiency of its processing units and minimizes unnecessary data movement overhead.

When benchmarked against the state-of-the-art Simba accelerator, AxLaM achieves a remarkable 9-fold reduction in energy use and a 58% decrease in area, underscoring its suitability for deployment in resource-constrained edge environments.

DTATrans, described by Yang et al. [32], and DTQAtten, outlined by Yang et al. [33], represent hardware-software co-designed solutions aimed at efficient transformer-based LLM inference on edge devices. These approaches leverage dynamic mixed-precision quantization and a variable-speed systolic array (VSSA) architecture to achieve exceptional performance and energy efficiency. Central to their design is a dynamic mixed-precision quantization scheme that adjusts the precision of model weights and activations based on their significance and the computational capacity of the target device, striking an effective balance between accuracy and efficiency. The accelerators employ a variable-speed systolic array architecture, which dynamically tunes the processing speed and parallelism of matrix operations to match workload characteristics and available resources, thereby optimizing both performance and energy use. Furthermore, DTATrans and DTQAtten benefit from a tight integration of hardware and software, encompassing the compiler, runtime, and programming model. This co-design ensures efficient mapping and scheduling of LLM workloads onto the hardware, enhancing overall effectiveness. In terms of results, DTATrans delivers a 16.04-fold speedup and a 3.62-fold energy saving over the earlier Eyeriss accelerator, while DTQAtten achieves a 3.62-fold speedup and a 4.22-fold improvement in energy efficiency compared to the state-of-the-art SpAtten accelerator, highlighting their significant contributions to edge-based LLM inference.

Figure 5 compares the performance and energy efficiency of the three hardware acceleration solutions, highlighting their effectiveness in enabling efficient LLM deployment on edge devices.

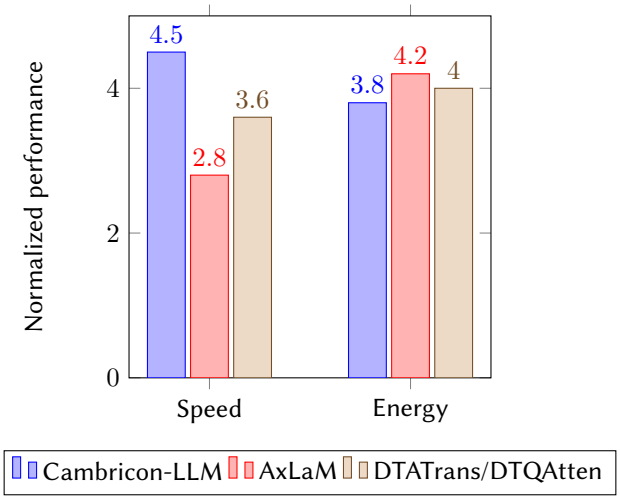

Figure 5: Comparison of hardware acceleration solutions in terms of speed and energy efficiency.

As shown in figure 5, all three hardware acceleration solutions achieve significant improvements in speed and energy efficiency compared to baseline edge devices without specialized accelerators. Cambricon-LLM demonstrates the highest speedup, while AxLaM and DTATrans/DTQAtten exhibit better energy efficiency. The choice of hardware solution depends on the specific requirements and constraints of the target application and deployment scenario, such as the model size, latency, and power budget.

4.2. Emerging trends and future directions

The domain of hardware acceleration for deploying large language models on edge devices is advancing at a brisk pace, unveiling a host of emerging trends and future research avenues aimed at enhancing the performance, efficiency, and scalability of these solutions. One notable direction involves the melding of diverse specialized accelerators – such as neural processing units, graphics processing units, and field-programmable gate arrays – into a single chip or package. This heterogeneous integration, as

explored in [34], facilitates the efficient handling of varied workload types and adapts seamlessly to the dynamic demands of edge LLM applications. Another promising development centres on in-memory computing, where innovative memory technologies like non-volatile memory and computational RAM are harnessed to perform computations directly within the memory itself. This approach, detailed in [19], significantly cuts down on data movement overhead between processing units and the memory hierarchy, boosting overall efficiency.

Equally compelling is the focus on exploiting sparsity within LLM structures. Researchers are delving into sparsity-aware hardware architectures and dataflows that capitalize on the inherently sparse nature of these models, thereby reducing both computational and memory demands during inference, as discussed by Bhardwaj et al. [8]. Alongside this, the concept of adaptive precision is gaining traction. This involves crafting accelerators capable of dynamically adjusting computational precision based on workload specifics and desired accuracy levels, a strategy outlined in [18] that minimizes energy use while maximizing system performance. Furthermore, the synergy of hardware and software through co-design is proving vital. By jointly optimizing the hardware architecture, software stack, and algorithms, this approach – highlighted in [32] – tailors solutions to the unique constraints and characteristics of target devices and applications, ensuring a more cohesive deployment process.

These evolving trends and forward-looking directions underscore the importance of a comprehensive strategy for hardware acceleration in edge LLM deployment. They emphasize the intricate interplay among hardware, software, and application layers, suggesting that a holistic perspective is essential for progress. By embracing these advancements, future edge devices stand poised to support increasingly sophisticated and efficient LLM-based applications, opening the door to novel use cases and enriched user experiences.

5. Applications and real-world systems

The deployment of large language models on edge devices has enabled a wide range of applications and real-world systems across various domains, such as IoT, personalized services, and multi-modal interaction [11, 35, 36]. These applications leverage the capabilities of edge-based LLMs to provide intelligent, responsive, and context-aware services to users while preserving privacy and reducing latency.

5.1. IoT and smart city applications

The integration of LLMs with IoT devices and smart city infrastructure has enabled the development of intelligent and adaptive systems that can process and analyze real-time data streams, providing actionable insights and optimized decision-making [24, 37]. Table 6 summarizes representative IoT and smart city applications of edge-based LLMs.

Table 6
Representative IoT and smart city applications of edge-based LLMs.

Application	Key features	Impact	References
Traffic forecasting	Spatio-temporal modeling, edge-cloud collaboration	Improved traffic management	Rong et al. [38]
Anomaly detection	Unsupervised learning, real-time processing	Enhanced system reliability	Zhang and Shi [37]

Edge-based LLMs have been applied to traffic forecasting and management systems to enable accurate and timely prediction of traffic conditions, optimizing the use of transportation infrastructure and reducing congestion [38]. These systems leverage the spatio-temporal modelling capabilities of LLMs to capture the complex dependencies between traffic flow, weather conditions, and road network topology, adapting to the dynamic and heterogeneous nature of urban environments.

For example, Rong et al. [38] propose a lightweight spatio-temporal generative LLM (LSGLLM) for large-scale traffic flow forecasting, which is deployed on edge devices and collaborates with cloud servers to process and analyze real-time traffic data efficiently. The LSGLLM model achieves superior performance compared to traditional baselines, demonstrating the effectiveness of edge-based LLMs in traffic management applications.

Edge-based LLMs have also been employed in anomaly detection and predictive maintenance systems for IoT and industrial applications, enabling the early identification of potential faults and the optimization of maintenance schedules [37]. These systems utilize the unsupervised learning capabilities of LLMs to model the normal behaviour of IoT devices and industrial equipment, detecting deviations and anomalies in real-time and triggering appropriate actions.

For instance, Zhang and Shi [37] propose a blockchain-based edge intelligence framework that integrates large AI models and IoT devices for anomaly detection and predictive maintenance in smart city applications. The framework leverages the distributed and secure nature of blockchain technology to enable the collaborative training and inference of LLMs across multiple edge devices, ensuring the privacy and integrity of the data and the models.

Figure 6 illustrates the architecture of a typical IoT and smart city application of edge-based LLMs, highlighting the key components and the data flow between the edge devices and the cloud servers.

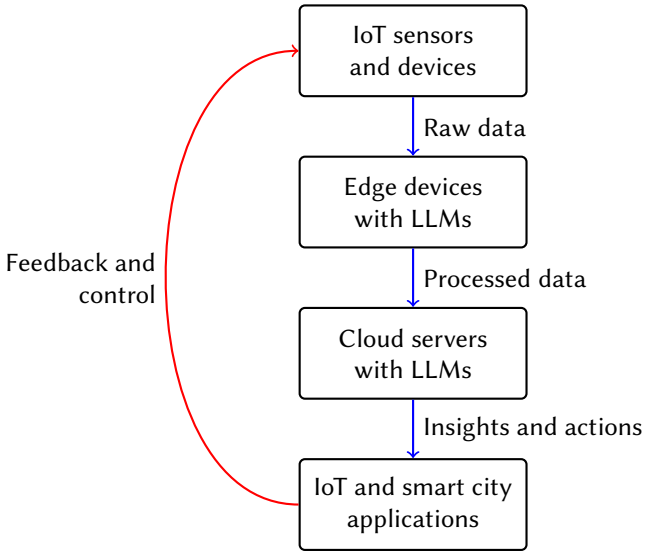

Figure 6: Architecture of a typical IoT and smart city application of edge-based LLMs.

As shown in figure 6, IoT sensors and devices collect raw data from the environment and send it to the edge devices for real-time processing and analysis using LLMs. The edge devices extract relevant features and patterns from the data and transmit the processed information to the cloud servers for further analysis and decision-making. The cloud servers utilize more powerful LLMs to generate insights and actionable recommendations, which are then fed back to the IoT and smart city applications for implementation. The applications, in turn, provide feedback and control signals to the sensors and devices, closing the loop and enabling adaptive and intelligent behaviour.

5.2. Personalized services and human-machine interaction

Edge-based LLMs have also been applied to personalized services and human-machine interaction applications, enabling the development of intelligent and context-aware systems that can understand and respond to user needs and preferences in real-time [35, 36]. Table 7 summarizes representative personalized services and human-machine interaction applications of edge-based LLMs.

Edge-based LLMs have been leveraged to develop intelligent personal assistants that can understand and respond to user queries and commands in natural language, providing personalized and context-

Table 7
Representative personalized services and human-machine interaction applications of edge-based LLMs.

Application	Key features	Impact	References
Intelligent personal assistants	Multi-modal interaction, context awareness	Enhanced user experience	Shen et al. [11]
Personalized recommendations	User profiling, collaborative filtering	Improved service quality	Piccialli et al. [39]

aware services [11]. These assistants integrate multi-modal interaction capabilities, such as speech recognition, computer vision, and natural language processing, to enable seamless and intuitive human-machine communication.

For example, Shen et al. [11] propose an edge-based autonomous AI system that leverages the capabilities of LLMs to provide high-quality, low-latency, and privacy-preserving personal assistant services. The system utilizes a combination of on-device LLMs and cloud-based models to process user queries and generate appropriate responses, adapting to the user's context and preferences.

Edge-based LLMs have also been applied to the personalized recommendation and content generation systems, enabling the delivery of tailored and engaging experiences to users based on their interests and behaviour [39]. These systems leverage the language understanding and generation capabilities of LLMs to create user profiles, analyze user feedback, and generate personalized recommendations and content.

For instance, Piccialli et al. [39] propose a federated and edge learning framework for LLMs that enables the collaborative training and inference of recommendation models across multiple edge devices while preserving user privacy. The framework utilizes techniques such as differential privacy and secure multi-party computation to ensure the confidentiality of user data and the integrity of the models.

Figure 7 illustrates the workflow of a typical personalized service or human-machine interaction application of edge-based LLMs, highlighting the key steps and the interaction between the user and the system.

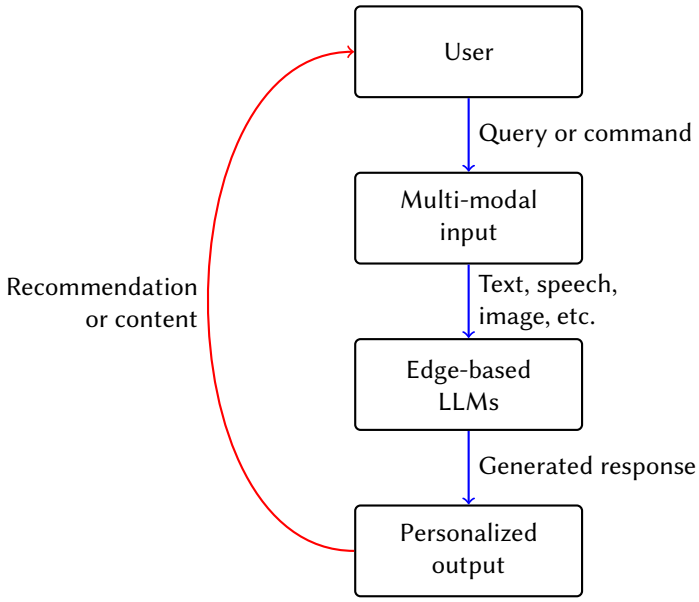

Figure 7: Workflow of a typical personalized service or human-machine interaction application of edge-based LLMs.

As shown in figure 7, the user initiates the interaction by providing a query or command to the system, which can be in the form of text, speech, image, or other modalities. The edge-based LLMs process the input and generate a personalized response, which is then presented to the user as a recommendation

or a piece of content. The user can provide feedback on the output, which is used by the system to refine the user profile and improve the quality of future recommendations and interactions.

5.3. Multi-modal edge intelligence

The integration of edge-based LLMs with other AI domains, such as computer vision, speech recognition, and robotics, has enabled the development of multi-modal edge intelligence applications that can process and analyze heterogeneous data streams in real-time [40, 41]. These applications leverage the complementary strengths of different AI technologies to provide more comprehensive and accurate insights and actions, enhancing the capabilities of edge devices and systems. Table 8 summarizes representative multi-modal edge intelligence applications of edge-based LLMs.

Table 8
Representative multi-modal edge intelligence applications of edge-based LLMs.

Application	Key features	Impact	References
Robotics and autonomous systems	Multi-modal perception, natural language interaction	Enhanced robot capabilities	Kawaharazuka et al. [42]
Computer vision and video analytics	Cross-modal learning, real-time processing	Improved scene understanding	Xu et al. [43]

Edge-based LLMs have been integrated with robotics and autonomous systems to enable more natural and intuitive human-robot interaction, as well as more robust and adaptive robot behaviour [42]. These systems leverage the language understanding and generation capabilities of LLMs to process and respond to human commands and queries while also utilizing the perception and action capabilities of robots to perform tasks in the physical world.

For example, Kawaharazuka et al. [42] propose a framework for applying pre-trained vision-language models to various recognition behaviours in robotic applications, enabling robots to understand and respond to visual and linguistic cues in real-world environments. The framework utilizes techniques such as zero-shot learning and prompt engineering to adapt the pre-trained models to specific tasks and domains without requiring extensive fine-tuning or data collection.

Edge-based LLMs have also been combined with computer vision and video analytics techniques to enable more accurate and efficient scene understanding and object recognition in real-time [43]. These applications leverage the cross-modal learning capabilities of LLMs to process and analyze visual and textual data streams simultaneously, extracting relevant features and generating semantic descriptions of the observed scenes.

For instance, Xu et al. [43] propose a benchmark suite for evaluating the performance of multi-modal deep neural networks (DNNs) in edge computing environments, focusing on the hardware and software implications of deploying these models on resource-constrained devices. The benchmark includes a set of representative computer vision and video analytics tasks, such as object detection, image captioning, and video summarization, and provides insights into the trade-offs between accuracy, latency, and energy efficiency of different multi-modal DNN architectures and optimization techniques.

Figure 8 illustrates the architecture of a typical multi-modal edge intelligence application of edge-based LLMs, highlighting the integration of different AI technologies and the flow of data and control between them.

As shown in figure 8, multi-modal sensors, such as cameras, microphones, and tactile sensors, collect data from the environment and send it to the respective AI modules for processing. The computer vision module extracts visual features and detects objects of interest, while the speech recognition module transcribes and interprets spoken commands and queries. The robotics and control module processes the sensor data and generates appropriate actions and behaviours for the robot. The edge-based LLMs integrate the outputs of the individual AI modules and generate a cohesive and semantically meaningful representation of the scene, which is then used to guide the robot's actions and interactions. The multi-modal output, such as natural language descriptions, visual explanations, and motor commands,

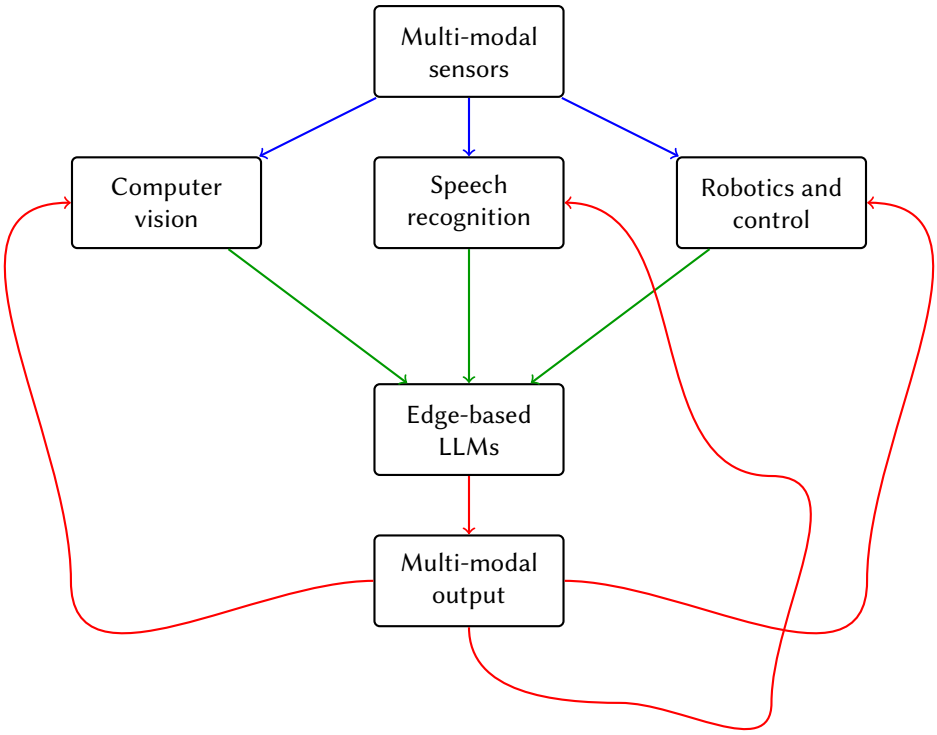

Figure 8: Architecture of a typical multi-modal edge intelligence application of edge-based LLMs.

is fed back to the AI modules for further processing and refinement, creating a closed-loop system that can adapt to dynamic and unstructured environments.

5.4. Case studies and real-world deployments

To demonstrate the practical impact and potential of edge-based LLMs, this section presents several case studies and real-world deployments of these technologies across different domains and applications. Table 9 summarizes the key features and outcomes of each case study.

Table 9
Case studies and real-world deployments of edge-based LLMs.

Case study	Domain	Key features	Outcomes
Autonomous edge AI systems [11]	Personal assistants	Multi-modal interaction, privacy-preserving	Enhanced user experience
LLM-powered smartphones [36]	Mobile devices	On-device inference, energy efficiency	Improved functionality and performance
Personalized digital avatars [35]	Human-computer interaction	Realistic appearance and voice, low latency	Engaging and immersive interactions

Shen et al. [11] present an autonomous edge AI system that uses LLMs to provide intelligent and personalized services to users, such as voice assistants, recommendation systems, and content generation. The system employs a hierarchical architecture that combines on-device LLMs for low-latency inference and privacy-preserving data processing with cloud-based models for more complex and computationally intensive tasks. The system also incorporates techniques such as federated learning and differential privacy to enable collaborative model training and adaptation across multiple edge devices while ensuring the security and confidentiality of user data.

The autonomous edge AI system has been deployed in various real-world scenarios, such as smart homes, connected vehicles, and personal robotics, demonstrating significant improvements in user

experience, service quality, and operational efficiency compared to traditional cloud-based solutions. The system has also been shown to reduce the energy consumption and network bandwidth requirements of edge devices, making it suitable for resource-constrained environments.

Wu et al. [36] investigate the integration of LLMs into smartphones to enable advanced functionality and improved performance for mobile users. The authors develop a framework for optimizing the deployment of LLMs on mobile devices, considering factors such as model compression, quantization, and hardware acceleration. The framework also includes a runtime system that dynamically adapts the inference process based on the available resources and the user's context, ensuring optimal performance and energy efficiency.

LLM-powered smartphones have been evaluated in terms of their natural language processing capabilities, such as text classification, language translation, and question answering, as well as their impact on the user experience and battery life. The results show that the optimized LLMs can achieve comparable accuracy to cloud-based models while significantly reducing the devices' latency and energy consumption. The LLM-powered smartphones have also been shown to enable new applications and services, such as on-device virtual assistants, real-time language translation, and personalized content recommendations.

Basit and Shafique [35] propose a multi-modal LLM-based framework for creating personalized digital avatars that can engage in natural and expressive interactions with users. The framework integrates LLMs for natural language processing, deep learning models for speech synthesis and recognition, and computer vision techniques for generating realistic facial expressions and gestures. The digital avatars are designed to run on edge devices, such as smartphones and smart speakers, providing low-latency and privacy-preserving interactions.

Personalized digital avatars have been evaluated in terms of their naturalness, expressiveness, and user engagement using both objective metrics and subjective user studies. The results show that the avatars can generate highly realistic and context-appropriate responses while also adapting to the user's preferences and emotions. The digital avatars have been deployed in various applications, such as virtual customer service agents, personal tutors, and social companions, demonstrating their potential to enhance the user experience and create more engaging and immersive interactions.

These case studies and real-world deployments highlight the diversity and impact of edge-based LLMs across different domains and applications. They also demonstrate the practical feasibility and benefits of deploying these technologies on resource-constrained devices, paving the way for more intelligent, responsive, and user-centric edge computing systems.

6. Challenges, opportunities, and future directions

Despite the significant advancements and promising applications of edge-based LLMs, several challenges and opportunities remain that need to be addressed to fully realize their potential.

6.1. Resource constraints and efficiency optimization

One of the main challenges in deploying LLMs on edge devices is the limited computational resources and energy budget of these devices [8]. To address this challenge, there is a need for more efficient and adaptive optimization techniques that can dynamically adjust the model architecture, hyperparameters, and deployment strategy based on the available resources and the target task. This includes the development of more advanced compression methods, such as quantization-aware training [18], network pruning [14], and knowledge distillation [20], as well as the exploration of novel architectures and learning paradigms, such as mixture-of-experts [44] and meta-learning [45].

Another important direction is the design of more efficient hardware accelerators and platforms for edge-based LLMs, considering the unique characteristics and requirements of these models [19]. This includes the development of specialized processing units, such as tensor processing units (TPUs) [46] and neural processing units (NPUs) [47], as well as the optimization of memory hierarchies and interconnects for fast and low-power data movement. Additionally, the co-design of hardware and

software components, such as compilers, runtime systems, and frameworks, can enable more seamless and efficient deployment of LLMs on edge devices [30].

6.2. Privacy, security, and trustworthiness

Another critical challenge in edge-based LLMs is ensuring the privacy, security, and trustworthiness of these systems, especially when dealing with sensitive user data and interactions [7]. To address this challenge, there is a need for more robust and scalable privacy-preserving techniques, such as federated learning [39], differential privacy [48], and homomorphic encryption [49], that can enable the collaborative training and inference of LLMs across multiple edge devices, without compromising the confidentiality and integrity of the data and the models.

Moreover, the development of more secure and resilient architectures and protocols for edge-based LLMs is crucial to prevent unauthorized access, tampering, and attacks on these systems [5]. This includes the use of trusted execution environments (TEEs) [50], blockchain-based authentication and access control [37], and anomaly detection and mitigation techniques [28]. Additionally, the incorporation of explainable and interpretable AI techniques [51] can enhance the transparency and accountability of edge-based LLMs, enabling users to understand and trust the decisions and actions of these systems.

6.3. Domain-specific adaptation and customization

Another important challenge and opportunity in edge-based LLMs is the adaptation and customization of these models to specific domains and applications [11]. While pre-trained LLMs can provide a good starting point for many tasks, they often require fine-tuning and domain-specific optimization to achieve optimal performance and user experience. This includes the incorporation of domain knowledge and constraints into the model architecture and training process, as well as the development of more efficient and effective transfer learning techniques [52].

Moreover, the design of more modular and composable LLMs that can be easily adapted and extended to new tasks and domains is an important research direction [53]. This includes the development of plug-and-play modules, such as adapters [21] and prefix-tuning [54], that can be seamlessly integrated into existing LLMs to enable fast and efficient adaptation to new requirements and preferences. Additionally, the exploration of more interactive and collaborative learning paradigms, such as active learning [55] and human-in-the-loop learning [56], can enable the continuous improvement and customization of edge-based LLMs based on user feedback and real-world usage patterns.

6.4. Scalability and interoperability in heterogeneous environments

Another challenge and opportunity in edge-based LLMs is ensuring their scalability and interoperability in heterogeneous and dynamic environments [9]. Edge computing systems often involve a large number of diverse devices, protocols, and platforms, which can hinder the seamless deployment and coordination of LLMs across these systems. To address this challenge, there is a need for more flexible and adaptive middleware and frameworks that can abstract away the underlying heterogeneity and enable the efficient and reliable communication and synchronization of LLMs across different edge devices and networks.

Moreover, the development of standardized interfaces and protocols for edge-based LLMs is crucial to enable their interoperability and compatibility with existing tools and services [6]. This includes the design of common architectures, such as the Open Neural Network Exchange (ONNX) [57], and the adoption of open standards, such as the Edge Computing Reference Architecture (ECRA) [58], to facilitate the integration and deployment of LLMs in edge computing environments. Additionally, the exploration of more decentralized and self-organizing architectures, such as peer-to-peer networks [59] and multi-agent systems [60], can enable the scalable and resilient coordination of edge-based LLMs in large-scale and dynamic environments.

6.5. Emerging applications and future research directions

Finally, there are many emerging applications and future research directions that can further advance the field of edge-based LLMs and unlock new opportunities for innovation and impact. Some of these directions include:

1. The integration of edge-based LLMs with other AI technologies, such as reinforcement learning [61], graph neural networks [62], and knowledge graphs [63], to enable more intelligent and context-aware decision making and reasoning in edge computing environments.
2. The exploration of edge-based LLMs for multi-modal and cross-lingual applications, such as image captioning [64], video summarization [65], and machine translation [66], to enable more natural and expressive interactions between humans and edge devices.
3. The development of edge-based LLMs for mission-critical and safety-critical applications, such as autonomous driving [67], industrial control systems [68], and healthcare monitoring [69], to ensure the reliability, security, and performance of these systems in real-world environments.
4. The investigation of edge-based LLMs for sustainable and green computing, considering the energy efficiency, carbon footprint, and environmental impact of these systems [70], and exploring techniques such as energy harvesting [71], workload consolidation [72], and renewable energy integration [73] to reduce their ecological footprint.
5. The study of the social, economic, and ethical implications of edge-based LLMs, including aspects such as fairness, accountability, transparency, and explainability [74], and the development of responsible AI principles and guidelines [75] to ensure the beneficial and trustworthy deployment of these technologies in society.

These emerging applications and future research directions highlight the vast potential and impact of edge-based LLMs in shaping the future of intelligent and sustainable computing systems.

7. Conclusion

This survey has explored the dynamic and rapidly advancing field of edge-based large language models (LLMs), a frontier that bridges the power of advanced AI with the constraints of resource-limited edge devices. We have examined the techniques – such as quantization, pruning, and knowledge distillation – that enable efficient LLM deployment alongside frameworks like TinyAgent, MNN-LLM, and h2oGPT that facilitate practical implementation. Edge-cloud collaborative architectures, including EdgeShard, Edge-LLM, and PAC, demonstrate how hybrid systems can overcome computational bottlenecks, while hardware innovations like Cambricon-LLM, AxLaM, and DTATrans/DTQAtten push the boundaries of performance and energy efficiency. These advancements have unlocked a diverse array of applications, from IoT-driven smart cities and personalized assistants to multi-modal edge intelligence, as illustrated by real-world deployments such as LLM-powered smartphones and autonomous AI systems.

The significance of edge-based LLMs extends beyond technical achievements; they represent a paradigm shift in how intelligent computing is delivered. By enabling low-latency, privacy-preserving, and context-aware AI at the edge, these models democratize access to cutting-edge capabilities, empowering users in resource-constrained environments – whether in remote areas with limited connectivity or urban settings demanding real-time responsiveness. The survey ties together these threads – techniques, frameworks, hardware, and applications – into a cohesive narrative: edge-based LLMs are not merely an adaptation of existing technology but a foundational step toward ubiquitous, sustainable intelligence. For instance, model compression and hardware acceleration address resource constraints, while collaborative frameworks and privacy-preserving techniques ensure scalability and trust, collectively paving the way for innovative applications that redefine human-machine interaction.

Looking forward, the future of edge-based LLMs is both promising and demanding. Continued innovation is needed to address persistent challenges, such as optimizing resource efficiency, enhancing privacy and security, and achieving seamless scalability across heterogeneous environments. Specific research directions include developing adaptive, modular LLM architectures for domain-specific

customization, integrating multi-modal reasoning for richer interactions, and exploring sustainable computing paradigms to minimize environmental impact. These efforts will unlock transformative possibilities – imagine autonomous systems reasoning in real-time, personalized services adapting instantly to user needs or green edge AI reducing the carbon footprint of intelligent devices.

This survey serves as both a comprehensive resource and a call to action for researchers and practitioners at the confluence of LLMs and edge computing. The advancements chronicled here are not endpoints but stepping stones toward a future where edge-based LLMs become integral to everyday life – secure, efficient, and universally accessible. We conclude with a bold vision: edge-based LLMs have the potential to reshape the landscape of computing, bringing intelligence closer to users than ever before and fostering a world where AI is not just powerful but personal, pervasive, and profoundly impactful.

Declaration on Generative AI: During the preparation of this work, the authors used Scopus AI to generate a literature review, Claude 3 Opus to draft content, Grok 3 for abstract drafting, improved writing style and citation management, and Grammarly for grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] A. V. Slobodianiuk, S. O. Semerikov, Advances in neural text generation: A systematic review (2022-2024), CEUR Workshop Proceedings 3917 (2025) 332–361.

[2] R. O. Liashenko, S. O. Semerikov, Bibliometric analysis and experimental assessment of chatbot training approaches, CEUR Workshop Proceedings 3917 (2025) 199–225.

[3] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: J. Burstein, C. Doran, 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, 2019, pp. 4171–4186. doi:10.18653/V1/N19-1423.

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

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

[6] F. Cai, D. Yuan, Z. Yang, L. Cui, 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, K. Wei (Eds.), Proceedings of the IEEE International Conference on Web Services, ICWS, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 799–809. doi:10.1109/ICWS62655.2024.00099.

[7] Q. Li, J. Wen, H. Jin, 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 (2024) 179. doi:10.1145/3699760.

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

[9] M. Zhang, X. Shen, J. Cao, Z. Cui, S. Jiang, EdgeShard: Efficient LLM Inference via Collaborative Edge Computing, IEEE Internet of Things Journal (2024). doi:10.1109/JIOT.2024.3524255.

[10] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, 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, 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

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

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

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is All you Need, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, 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, 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[14] D. Qiao, X. Ao, Y. Liu, X. Chen, F. Song, Z. Qin, W. Jin, Tri-AFLLM: Resource-Efficient Adaptive Asynchronous Accelerated Federated LLMs, IEEE Transactions on Circuits and Systems for Video Technology (2024). doi:10.1109/TCSVT.2024.3519790.

[15] L. E. Erdogan, N. Lee, S. Jha, S. Kim, R. Tabrizi, S. Moon, C. Hooper, G. Anumanchipalli, K. Keutzer, A. Gholami, TinyAgent: Function Calling at the Edge, in: D. I. H. Farias, T. Hope, M. Li (Eds.), EMNLP 2024 - 2024 Conference on Empirical Methods in Natural Language Processing, Proceedings of System Demonstrations, Association for Computational Linguistics (ACL), 2024, pp. 80–88.

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

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

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

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

[20] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, CoRR abs/1910.01108 (2019). URL: http://arxiv.org/abs/1910.01108. arXiv:1910.01108.

[21] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-Efficient Transfer Learning for NLP, in: K. Chaudhuri, R. Salakhutdinov (Eds.),

Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 2790–2799. URL: http://proceedings.mlr.press/v97/houlsby19a.html.

[22] G. E. Hinton, O. Vinyals, J. Dean, Distilling the Knowledge in a Neural Network, CoRR abs/1503.02531 (2015). URL: http://arxiv.org/abs/1503.02531. arXiv:1503.02531.

[23] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, Q. Liu, TinyBERT: Distilling BERT for natural language understanding, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 4163–4174. doi:10.18653/v1/2020.findings-emnlp.372.

[24] X. Zhou, Q. Jia, Y. Hu, R. Xie, T. Huang, F. R. Yu, GenG: An LLM-Based Generic Time Series Data Generation Approach for Edge Intelligence via Cross-Domain Collaboration, in: IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2024, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 1–6. doi:10.1109/INFOCOMWKSHPS61880.2024.10620716.

[25] W. Zhao, W. Jing, Z. Lu, X. Wen, Edge and Terminal Cooperation Enabled LLM Deployment Optimization in Wireless Network, in: International Conference on Communications in China, ICCC Workshops 2024, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 220–225. doi:10.1109/ICCCWorkshops62562.2024.10693742.

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

[27] Z. Yao, Z. Tang, J. Lou, P. Shen, W. Jia, 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, K. Wei (Eds.), Proceedings of the IEEE International Conference on Web Services, ICWS, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 865–876. doi:10.1109/ICWS62655.2024.00105.

[28] N. Nazari, F. Xiang, C. Fang, H. M. Makrani, A. Puri, K. Patwari, H. Sayadi, S. Rafatirad, C.-N. Chuah, H. Homayoun, LLM-FIN: Large Language Models Fingerprinting Attack on Edge Devices, in: Proceedings - International Symposium on Quality Electronic Design, ISQED, IEEE Computer Society, 2024, pp. 1–6. doi:10.1109/ISQED60706.2024.10528736.

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

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

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

[32] T. Yang, F. Ma, X. Li, F. Liu, Y. Zhao, Z. He, L. Jiang, 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 (2023) 509–520. doi:10.1109/TCAD.2022.3181541.

[33] T. Yang, D. Li, Z. Song, Y. Zhao, F. Liu, Z. Wang, Z. He, L. Jiang, DTQAtten: Leveraging Dynamic Token-based Quantization for Efficient Attention Architecture, in: C. Bolchini, I. Verbauwhede, 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., 2022, pp. 700–705. doi:10.23919/DATE54114.2022.9774692.

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

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

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

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

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

[39] F. Piccialli, D. Chiaro, P. Qi, V. Bellandi, E. Damiani, Federated and edge learning for large language models, Information Fusion 117 (2025) 102840. doi:10.1016/j.inffus.2024.102840.

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

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

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

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

[44] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton, J. Dean, Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017. URL: https://openreview.net/forum?id=B1ckMDqlg.

[45] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, JMLR.org, 2017, p. 1126–1135.

[46] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-l. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, D. Killebrew, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn,

G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, D. H. Yoon, In-Datacenter Performance Analysis of a Tensor Processing Unit, SIGARCH Comput. Archit. News 45 (2017) 1–12. doi:10.1145/3140659.3080246.

[47] Y. Xue, Y. Liu, J. Huang, System Virtualization for Neural Processing Units, in: Proceedings of the 19th Workshop on Hot Topics in Operating Systems, HOTOS '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 80–86. doi:10.1145/3593856.3595912.

[48] C. Dwork, A. Roth, The Algorithmic Foundations of Differential Privacy, Foundations and Trends in Theoretical Computer Science 9 (2014) 211–407. doi:10.1561/0400000042.

[49] D. Kim, Y. Lee, S. Cheon, H. Choi, J. Lee, H. Youm, D. Lee, H. Kim, Privacy Set: Privacy-Authority-Aware Compiler for Homomorphic Encryption on Edge-Cloud System, IEEE Internet Things J. 11 (2024) 35167–35184. doi:10.1109/JIOT.2024.3437356.

[50] M. Sabt, M. Achemlal, A. Bouabdallah, Trusted Execution Environment: What It is, and What It is Not, in: 2015 IEEE Trustcom/BigDataSE/ISPA, volume 1, 2015, pp. 57–64. doi:10.1109/Trustcom.2015.357.

[51] P. Dubey, M. Kumar, Integrating Explainable AI with Federated Learning for Next-Generation IoT: A comprehensive review and prospective insights, Computer Science Review 56 (2025) 100697. doi:10.1016/J.COSREV.2024.100697.

[52] A. Petrella, M. Miozzo, P. Dini, Mobile Traffic Prediction at the Edge Through Distributed and Deep Transfer Learning, IEEE Access 12 (2024) 191288–191303. doi:10.1109/ACCESS.2024.3518483.

[53] B. Thomas, S. Kessler, S. Karout, Efficient Adapter Transfer of Self-Supervised Speech Models for Automatic Speech Recognition, in: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 7102–7106. doi:10.1109/ICASSP43922.2022.9746223.

[54] L. Falissard, S. Affeldt, M. Nadif, Attentive Perturbation: Extending Prefix Tuning to Large Language Models Inner Representations, in: G. Nicosia, V. Ojha, E. L. Malfa, G. L. Malfa, P. M. Pardalos, R. Umeton (Eds.), Machine Learning, Optimization, and Data Science - 9th International Conference, LOD 2023, Grasmere, UK, September 22-26, 2023, Revised Selected Papers, Part I, volume 14505 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 488–496. doi:10.1007/978-3-031-53969-5_36.

[55] B. Yuan, Y. Chen, Y. Zhang, W. Jiang, Hide and Seek in Noise Labels: Noise-Robust Collaborative Active Learning with LLMs-Powered Assistance, in: L. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, Association for Computational Linguistics, 2024, pp. 10977–11011. doi:10.18653/V1/2024.ACL-LONG.592.

[56] Q. Zhang, C. Xu, J. Li, Y. Sun, J. Bao, D. Zhang, LLM-TSFD: An industrial time series human-in-the-loop fault diagnosis method based on a large language model, Expert Syst. Appl. 264 (2025). doi:10.1016/j.eswa.2024.125861.

[57] M. Garofalo, M. Colosi, A. Catalfamo, M. Villari, Web-Centric Federated Learning over the Cloud-Edge Continuum Leveraging ONNX and WASM, in: IEEE Symposium on Computers and Communications, ISCC 2024, Paris, France, June 26-29, 2024, IEEE, 2024, pp. 1–7. doi:10.1109/ISCC61673.2024.10733614.

[58] I. D. Martinez-Casanueva, L. Bellido, C. M. Lentisco, D. Fernández, An Initial Approach to a Multi-access Edge Computing Reference Architecture Implementation Using Kubernetes, in: H. Gao, R. J. D. Barroso, S. Pang, R. Li (Eds.), Broadband Communications, Networks, and Systems - 11th EAI International Conference, BROADNETS 2020, Qingdao, China, December 11-12, 2020, Proceedings, volume 355 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer, 2020, pp. 185–193. doi:10.1007/978-3-030-68737-3_13.

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

[60] F. Zhu, F. Huang, Y. Yu, G. Liu, T. Huang, Task Offloading with LLM-Enhanced Multi-Agent

Reinforcement Learning in UAV-Assisted Edge Computing, Sensors 25 (2025) 175. doi:10.3390/s25010175.

[61] M. Xu, D. Niyato, C. G. Brinton, Serving Long-Context LLMs at the Mobile Edge: Test-Time Reinforcement Learning-based Model Caching and Inference Offloading, CoRR abs/2501.14205 (2025). doi:10.48550/ARXIV.2501.14205. arXiv:2501.14205.

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

[63] M. Yang, Y. Yang, P. Jiang, 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 (2024) 2584–2605. doi:10.1080/00207543.2023.2219345.

[64] N. Wang, J. Xie, H. Luo, Q. Cheng, J. Wu, M. Jia, L. Li, Efficient Image Captioning for Edge Devices, in: B. Williams, Y. Chen, 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, 2023, pp. 2608–2616. doi:10.1609/AAAI.V37I2.25359.

[65] Y. Wang, Y. Dong, S. Guo, Y. Yang, X. Liao, Latency-Aware Adaptive Video Summarization for Mobile Edge Clouds, IEEE Trans. Multim. 22 (2020) 1193–1207. doi:10.1109/TMM.2019.2939753.

[66] R. Liashenko, S. Semerikov, The Determination and Visualisation of Key Concepts Related to the Training of Chatbots, in: E. Faure, Y. Tryus, T. Vartiainen, O. Danchenko, M. Bondarenko, C. Bazilo, G. Zaspa (Eds.), Information Technology for Education, Science, and Technics, volume 222 of *Lecture Notes on Data Engineering and Communications Technologies*, Springer Nature Switzerland, Cham, 2024, pp. 111–126. doi:10.1007/978-3-031-71804-5_8.

[67] V. Mukovoz, T. Vakaliuk, S. Semerikov, Road Sign Recognition Using Convolutional Neural Networks, in: Information Technology for Education, Science, and Technics, volume 222 of *Lecture Notes on Data Engineering and Communications Technologies*, Springer Nature Switzerland, Cham, 2024, pp. 172–188. doi:10.1007/978-3-031-71804-5_12.

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

[69] S. Ji, X. Zheng, J. Sun, R. Chen, W. Gao, M. Srivastava, MindGuard: Towards Accessible and Sitgma-free Mental Health First Aid via Edge LLM, CoRR abs/2409.10064 (2024). doi:10.48550/ARXIV.2409.10064. arXiv:2409.10064.

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

[71] A. Khoshsirat, G. Perin, M. Rossi, Decentralized LLM inference over edge networks with energy harvesting, CoRR abs/2408.15907 (2024). doi:10.48550/ARXIV.2408.15907. arXiv:2408.15907.

[72] I. Mohiuddin, A. Almogren, Workload aware VM consolidation method in edge/cloud computing for iot applications, J. Parallel Distributed Comput. 123 (2019) 204–214. doi:10.1016/J.JPDC.2018.09.011.

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

[74] D. O. Hanchuk, S. O. Semerikov, Implementing MLOps practices for effective machine learning model deployment: A meta synthesis, CEUR Workshop Proceedings 3918 (2024) 329–337.

[75] D. O. Hanchuk, S. O. Semerikov, Automating machine learning: A meta-synthesis of MLOps tools, frameworks and architectures, CEUR Workshop Proceedings 3917 (2025) 362–414.