Edge-Enhanced Intelligence: A Comprehensive Survey of Large Language Models and Edge-Cloud Computing Synergy

Hui Li, *Student Member, IEEE,* Xiuhua Li, *Member, IEEE,* Chuan Sun, *Member, IEEE,* Xiaofei Wang, *Senior Member, IEEE,* Victor C. M. Leung, *Life Fellow, IEEE*

***Abstract*—Large language models (LLMs) (e.g., *ChatGPT*, *GPT-4* and *Sora*) have fundamentally transformed our daily lives, catalyzing breakthroughs in natural language processing, computer vision and revolutionizing human-computer interactions. However, their transformative potential is hindered by the immense computational resources required. To address this challenge, the integration of LLMs with edge-cloud computing infrastructure has become a focal point in advancing the capabilities of artificial intelligence. This survey comprehensively exploits the landscape of edge-enhanced intelligence, specifically focusing on the synergy between LLMs and edge-cloud computing. The paper delves into the evolution of LLMs, their architectural intricacies, and the computational challenges associated with deploying them in edge environments from the aspects of data, computing power, model training and inference, etc. Furthermore, we exploit the bidirectional symbiotic relationship between edge-cloud computing and LLMs from two key aspects: edge-cloud computing empowered LLMs, i.e., *Edge4LLMs* and LLMs driven edge-cloud computing, i.e., *LLMs4Edge*. After that, the survey examines the dynamic collaboration between edge-cloud computing and LLMs, highlighting their complementary roles in optimizing the efficiency, real-time, and scalability of intelligent applications. Through an extensive review of existing research and practical implementations, this survey offers insights into the current state of edge-enhanced intelligence, identifies key challenges, and proposes potential avenues for future research and development. This survey aims to provide a comprehensive understanding of the emerging paradigm of edge-enhanced intelligence for LLMs, fostering informed discussions and inspiring advancements in this field.**

***Index Terms*—Large language models, edge-cloud computing, wireless communication, edge intelligence, artificial intelligence.**

This work is supported in part by National Key R & D Program of China (Grants No. 2022YFE0125400), National NSFC (Grants No. 62372072, 62102053, 62072060, 92067206 and 61972222), Chongqing Research Program of Basic Research and Frontier Technology (Grant No. cstc2022ycjhbgzxm0058), Key Research Program of Chongqing Science & Technology Commission (Grant No. cstc2021jscx-dxwtBX0019), Science and Technology Plan Project of Chongqing Economic and Information Commission (Grant No. 2211R49R03), Haihe Lab of ITAI (Grant No. 22HHXCJC00002), the Natural Science Foundation of Chongqing, China (Grant No. CSTB2022NSCQ-MSX1104), the General Program of Chongqing Science & Technology Commission (Grant No. CSTB2022TIAD-GPX0017, CSTB2022TIAD-STX0006), Key Laboratory of Big Data Intelligent Computing, Chongqing University of Posts and Telecommunications (Grant No. BDIC-2023-B-003), Regional Innovation Cooperation Project of Sichuan Province (Grant No. 2023YFQ0028), Regional Science and Technology Innovation Cooperation Project of Chengdu City (Grant No. 2023-YF11-00023-HZ), and Guangdong Pearl River Talent Recruitment Program (Grants No. 2019ZT08X603 and 2019JC01X235). (Corresponding author: Xiuhua Li).

H. Li and X. Li are with the School of Big Data & Software Engineering, Chongqing University, Chongqing 401331, China (e-mail: h.li@cqu.edu.cn, lixiuhua1988@gmail.com).

C. Sun is with the School of Computer Science and Engineering, Nanyang Technological University, 639956 Singapore (e-mail: chuan.sun@ntu.edu.sg).

X. Wang is with the Tianjin Key Laboratory of Advanced Networking, School of Computer Science and Technology, Tianjin University, Tianin 300072, China (e-mail:xiaofeiwang@tju.edu.cn).

V. C. M. Leung is with the College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T1Z4, Canada (e-mail: vleung@ieee.org).

I. Introduction

A. Background

Artificial general intelligence (AGI) has always been a pursuit goal within the realms of science and technology. [1]–[3]. It represents an intelligent system that can comprehensively understand, learn and perform various tasks like humans. OpenAI is actively exploring various approaches to unlock the potential of achieving AGI [4]. The realization of AGI will bring profound changes to human society, and its potential impact covers almost all fields (e.g., health care, transportation, education) [5], [6]. However, the realization of AGI still faces great difficulties. First, human intelligence encompasses complex cognition, emotion, and creative thinking, which makes it extremely complex for machines to replicate. Secondly, realizing AGI requires machines to be able to flexibly learn and adapt in unknown environments, which requires overcoming the task-specific limitations of current narrow artificial intelligence (AI) [3].

Large language models (LLMs) play a pivotal role in promoting the development of AGI [7]–[9]. Recently, LLMs such as *ChatGPT*, *GPT-4* [10] and *Sora* have made significant progress in the fields of natural language processing (NLP), computer vision (CV) and video generation. The powerful language understanding and generation capabilities of LLMs provide machines with the basis for a wide range of learning in different fields. By combining LLMs with other techniques (e.g., Prompt learning or In-context learning), we can gradually address the task-specific challenge of narrow AI. This integrated approach takes a critical step towards the goal of comprehensive understanding and learning of various tasks by machines, bringing a more feasible prospect to the realization of AGI. Currently, we can access some typical LLMs through browsers or office software (e.g., *Hugging Face*, *New Bing* and *Microsoft* 365). However, as the model size increases, it also brings training and inference challenges. The huge amount of parameters leads to huge computing power requirements.

According to Tirias Research, 20% AI workloads offloaded from data centres through on-device and hybrid processing, the cost of infrastructure and operating will decline by $15 billion [11]. At the same time, a large amount of data needs to be processed, and the inference delay may be relatively long [12], which is unsuitable for some applications that require high real-time performance, such as autonomous driving or decision-making in emergencies. Therefore, in the pursuit of the performance of LLMs, we need to balance the potential advantages and practical challenges.

To cope with the challenges of real-time performance and inference delay brought by LLMs, the edge-cloud computing paradigm has become a potential solution that has attracted much attention. Edge-cloud computing pushes computing power resources closer to network edges (e.g., base stations (BSs)), thereby reducing data transmission delay [13], [14], which is critical for application scenarios that require fast response. Currently, the development of edge hardware [15]–[17], hardware accelerators [18] and other optimization technologies (model compression [19], [20] and quantification, etc.) has made it feasible to run LLMs on edge devices [21] (such as smartphones and internet-of-things (IoT) devices). On the other hand, by collaborative inference of LLMs at network edges, we can more effectively utilize the power of LLMs to provide intelligent decisions and service capability to edge devices. Therefore, on the road to AGI, LLMs and edge-cloud computing have formed a powerful synergistic relationship, jointly promoting the evolution of future AI technology.

Recently, some studies [22]–[28] have focused on the integration of LLMs and edge-cloud computing in different scenarios. The authors in [22] propose a novelty task offloading framework named *LAMBO* [22] that combines LLMs with mobile edge computing (MEC) networks for achieving edge-enhanced intelligence. These studies [23], [24] exploit the feasibility of bringing LLMs to edge computing (EC) networks and fine-tune LLMs through Federated Learning (FL), which offers valuable insights for future advancements in federated fine-tuning of LLMs on network edges. Wen *et al.* [25] design a mobile task automation system *AutoDroid* by leveraging LLMs to handle arbitrary tasks on Android applications in mobile devices. Additionally, this paper [28] proposes a memory-efficient algorithm *ModuLoRA* for LLMs with fine-tuning on consumer graphics processing units (GPUs). Consumer GPUs offer a distinct advantage at the edge of networks due to their widespread availability and seamless integration, making them easily accessible for a variety of edge applications. These innovation frameworks, systems or algorithms [22]–[28] have highlighted the feasibility and adaptability of running sophisticated models on edge devices.

Actually, the effective integration of edge-cloud computing and LLMs mainly presents two eye-catching forms, namely *LLMs4Edge* and *Edge4LLMs*, as depicted in Fig. 1. The LLM landscape spans text, image, code, and video domains, showcasing its versatility in comprehending and generating content across diverse modalities, as shown in Fig. 2. *LLMs4Edge* emphasizes the utilization of LLMs within edge-cloud networks and deploys them on edge servers/devices to achieve more intelligent real-time data processing and data generation

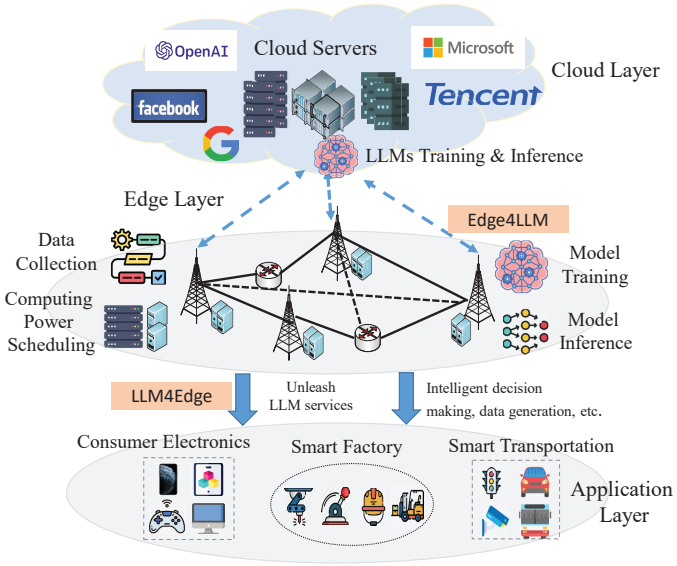


Fig. 1. Overview of *Edge4LLMs* and *LLMs4Edge*, including cloud layer, edge layer and application layer.

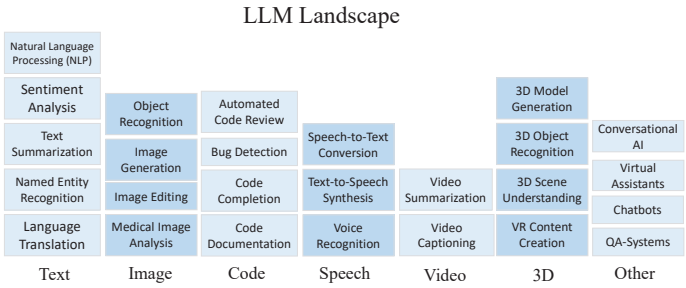


Fig. 2. LLM landscape.

service. It pushes intelligent decision-making and language understanding capabilities closer to edge users, providing more efficient and responsive language interactions for smartphones or IoT devices. This strategy is effective for enhancing the capabilities of edge devices such as smartphones and IoT devices, where instantaneous and intelligent language interactions are imperative. On the other hand, *Edge4LLMs* is used to leverage the potential of collaborative computing paradigm between edge and cloud to strategically optimize the training and inference efficiency of LLMs. Specifically, by migrating partial or entire models to network edges, *Edge4LLMs* can reduce the communication/computation delay while maintaining the utilization of idle powerful computing power.

Together, these integration forms pave the way for a seamless and efficient convergence of edge-cloud computing and LLMs, fostering a rich ecosystem of intelligent applications. This deep integration provides flexibility in adapting to diverse use cases and enhances the overall efficiency of intelligent systems, marking a significant stride towards the realization of a more intelligent and connected world. As the synergy between edge-cloud computing and LLMs continues to evolve, the possibilities for innovative applications across industries are boundless, heralding a new era in intelligent computing. Some important notations are listed in Table I.

B. Motivation of Edge-Cloud Computing and LLMs

In this subsection, we first discuss the limitations of LLMs, and then we discuss the benefits of edge-cloud computing. Finally, we explain the motivation for the integration of edge-cloud computing and LLMs in detail.

1) Limitations of LLMs: LLMs exhibit various limitations in terms of data, computing power resources, model training and model inference:

- Limited data. The performance of LLMs is highly dependent on large amounts of high-quality training data [29]. Nevertheless, obtaining and curating such datasets can pose significant challenges, particularly within a narrow and specialized domain. Limited data may cause the poor performance of LLMs in understanding certain contexts or specialized terminology [30].
- Insufficient computing power resources. Running LLMs requires huge computing resources, including high-performance hardware [31] and large-scale distributed computing systems. Many users may have difficulty acquiring the expensive hardware and infrastructure, restricting their capacity to harness the potential of LLMs.
- Time-consuming and costly training. The training process of LLMs may take massive time due to large parameters. The escalating demand for computing/storage resources has resulted in a substantial rise in training costs, rendering the training LLMs both intricate and expensive.
- High-latency inference. In real-world applications, the inference process of LLMs may suffer from significant delays [32], especially on edge devices with limited resources. This high latency can impact the performance of real-time applications, such as speech recognition or instant response from smart assistants.

2) Advantages of Edge-Cloud Computing: The advantages of edge-cloud computing are reflected in many aspects, including data collection close to data sources, efficient use of edge idle hardware resources, collaborative training on different edge nodes, and edge inference.

- Nearby data collection. Edge-cloud computing networks can achieve nearby data collection and processing closer to data sources [33], [34]. This advantage reduces the cost of data transmission, making the processing of real-time data more faster and efficient.
- Efficient edge resource utilization. Edge-cloud computing networks make full use of idle computing/storage resources of edge devices and avoid over-reliance on central cloud servers by intelligently allocating LLM tasks [35]. This efficient use of edge hardware resources not only improves model performance, but also reduces the burden on central cloud servers and overall resource overhead.
- Collaborative training. Edge-cloud computing networks support collaborative training on edge nodes to achieve distributed optimization of model parameters [36], [37]. This approach avoids transmitting large amounts of data to the central cloud servers for training, reduces communication overhead and protects user privacy [38].
- Edge inference. Edge-cloud computing enables real-time inference on edge devices without transmitting all data to central cloud servers [39], [40]. This real-time inference method reduces communication delays and is suitable for applications (e.g., intelligent monitoring and autonomous driving) with high real-time requirements.

TABLE I
SUMMARY OF IMPORTANT NOTATIONS

Notation	Definition
AGI	Artificial general intelligence
AI	Artificial intelligence
AIGC	Artificial intelligence generated content
BC	Blockchain
CPNs	Computing power networks
CV	Computer vision
DL	Deep learning
DNNs	Deep neural networks
EC	Edge computing
DRL	Deep reinforcement learning
FL	Federated learning
FPGA	Field-programmable gate array
FM	Foundation model
GPU	Graphics processing unit
IoT	Internet of thing
IoV	Internet of vehicles
KD	Knowledge distillation
LLMops	Operational processes of LLMs
LLMs	Large language models
MEC	Multi-Access/Mobile edge computing
ML	Machine learning
NLP	Natural language processing
PTQ	Post-training quantization
QAT	Quantization-aware training
QoE	Quality of experience
QoS	Quality of services
RLHF	Reinforcement learning from human feedback
SCSs	Sensor-cloud systems
SLA	Service level agreement
SPs	Service providers
TL	Transfer learning
TPU	Tensor processing unit
VEC	Vehicular edge computing

3) Integration of Edge-Cloud Computing and LLMs: Integrating edge-cloud computing and LLMs is an emerging technology trend. It aims to provide more efficient and flexible computing/storage resource support for LLMs by taking full advantage of EC and cloud computing. Fig. 3 shows the motivation for integration of edge-cloud computing and LLMs. This integration accelerates the training and inference process of LLMs and optimizes resource utilization.

Edge-cloud computing plays a key role in data management, computing power optimization, model training and inference [41] for LLMs. For data management, Training LLMs require large-scale, high-quality datasets. Currently, edge servers are generally closer to data sources and can achieve a lower data collection cost [42]. In terms of computing power resource optimization, there are a large number of scattered and idle computing power resources at network edges [43], [44]. Edge networks can effectively integrate these computing power resources. From the aspect of model training and inference, EC reduces communication delays and improves the performance of model inference by pushing computing resources closer to

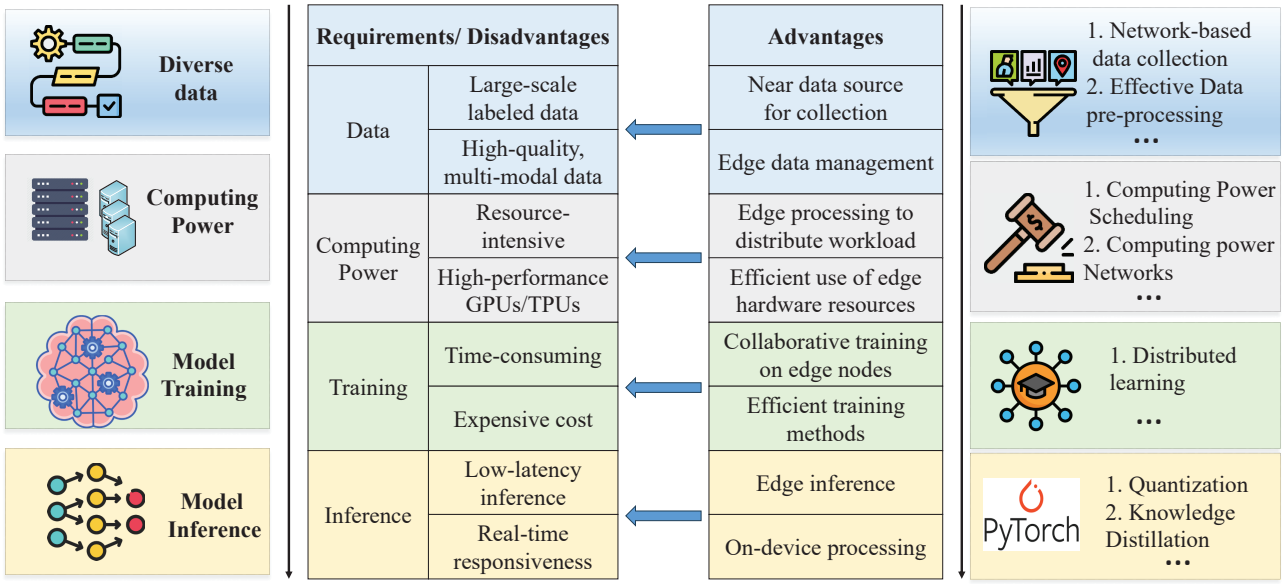


Fig. 3. Motivation for integration of edge-cloud computing and LLMs.

edge devices. Cloud computing provides more powerful computing and storage capabilities to support large-scale training tasks [45]. By intelligently allocating tasks between the edge and cloud servers, the system can achieve efficient resource utilization and reasonably distribute the computing load to different layers, thereby optimizing the efficiency of LLMs. This integration enables LLMs to better adapt to various application scenarios.

On the other hand, LLMs also provide huge advantages for edge-cloud networks. Specifically, LLMs can empower edge-cloud networks in terms of data generation, policy-making, and customized personalized services. LLMs have demonstrated excellent performance in NLP, image recognition and other tasks, providing powerful information processing and analysis capabilities for various application scenarios. By deploying lightweight LLMs on edge devices, localized real-time decision-making can be achieved and service dependencies on cloud servers can be reduced. LLMs serve as intelligence engines on edge devices to process real-time data, while working in conjunction with cloud servers to complete more complex and large-scale tasks. This bidirectional collaborative relationship enables edge-cloud networks to more comprehensively meet the needs of different applications and achieve a higher level of intelligence and flexibility. Therefore, it is attractive to investigate how edge-cloud computing and LLMs can be efficiently combined to enhance service performance.

C. Related Work and Our Tutorial

Recently, many studies have conducted research on EC, machine learning (ML), deep learning (DL) and LLMs. We listed some related surveys or tutorials in Table II. This survey [46] exploits EC as a crucial solution for network operators, detailing its integration into current mobile networks. It provides a comprehensive state-of-art study on diverse approaches to optimizing network resources and raises some open issues. This survey [47] exploits the application of ML/DL in MEC networks for efficient resource allocation. The authors provide tutorials on the advantages of ML/DL, discuss enabling technologies for ML/DL training and inference, and identify key research directions. For LLMs, the studies [48], [49] provide a comprehensive survey of AI-generated content (AIGC), focusing on the widespread use of large AI models like ChatGPT. It covers the working principles, security and privacy threats, state-of-the-art solutions, and future challenges in the AIGC paradigm. This survey [50] serves as recent advancements in LLMs. The authors cover key aspects of LLMs, including pre-training, adaptation tuning, utilization,

TABLE II
COMPARISON OF THE EXISTING SURVEYS

Ref	Topic	Main contribution
[46]	EC	Exploit EC as a crucial solution for network operators.
[47]	ML/DL	Examine ML/DL solutions for optimizing resource allocation at MEC networks.
[48], [49]	LLMs	Focus on operational principles, cutting-edge solutions, and future challenges of LLMs.
[50]	LLMs	Discuss the entire life cycle of LLMs and future direction.
[51]	DL+EC	Discuss application scenarios, practical implementation and enabling technologies to achieve edge intelligence, intelligent edge.
[52]	Distributed AI+EC	Introduce fundamental technologies and emphasize the benefits for supporting distributed AI.
[53]	Distributed AI+EC	Investigate state-of-the-art libraries, frameworks, learning paradigms, and experimental systems for distributed AI.
[54]	Edge for LLMs	Discuss 6G MEC architecture and some key techniques for efficient LLM deployment.
[55]	AIGC+EC	Discuss the life cycle of AIGC services, collaborative infrastructure and technologies supporting real-time AIGC services at mobile AIGC networks.
[56]	AIGC+EC	Offering a promising solution for seamless delivery AIGC services in wireless networks.
Our work	LLMs+EC	Exploit the opportunity, advantages and challenges of LLM for Edge, Edge for LLM.

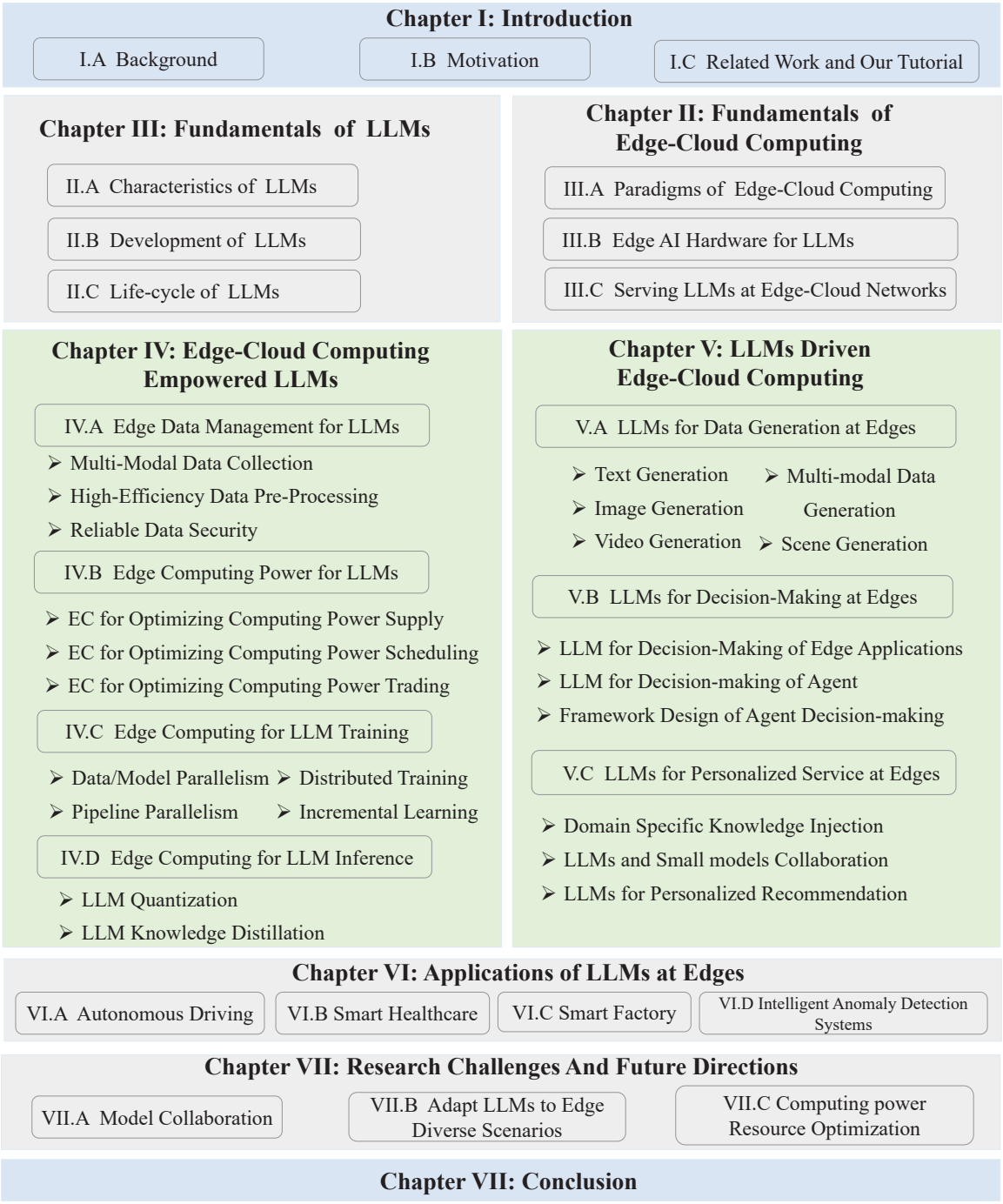


Fig. 4. Organization graph of this survey.

and capacity evaluation, offering an up-to-date resource for researchers and engineers, and showcasing the significant impact of LLMs on AI community.

With the advanced function of these technologies, an increasing number of studies have diligently explored the intricate relationships and collaborative potentials within EC and ML/DL. Currently, there have been many efforts that attempt to integrate EC, ML/DL and LLMs to compensate for each other's weak points. This comprehensive [51] survey exploits the convergence of EC and DL, outlining application scenarios, and detailing practical implementation methods and enabling technologies, such as customized EC frameworks for DL training and inference at network edges. Furthermore, in terms of distributed AI on end-edge-cloud computing networks, this survey [52] presents a comprehensive survey on distributed AI empowered by end-edge-cloud computing. The authors introduce fundamental technologies, outline various computing paradigms and emphasize the benefits of supporting distributed AI. Apart from this, in [53], the authors comprehensively investigate state-of-the-art libraries, frameworks, learning paradigms, and experimental systems for learning-based analytics, and exploit the landscape of distributed AI at

the edge-cloud environment.

The emergence of LLMs has captured the attention of researchers, sparking widespread interest and exploration. It has spurred investigations into seamlessly integrating these LLMs with EC. From the perspective of EC empowering LLMs, this article [54] envisions 6G MEC architecture for LLMs and discusses cutting-edge techniques like split learning, parameter-efficient fine-tuning, quantization, and parameter-sharing inference for efficient LLM deployment at network edges. This survey [55] exploits the deployment of artificial intelligence-generated content (AIGC) applications, such as ChatGPT and Dall-E, in mobile AIGC networks. The authors cover the life-cycle of AIGC services, collaborative infrastructure, and technologies to support real-time AIGC services for a comprehensive realization of mobile AIGC networks. This article [56] leverages device collaboration to optimize edge computation resource utilization, offering a promising solution to existing limitations in resource optimization and seamless delivery of AIGC services in wireless networks.

Different from these existing surveys, this survey comprehensively exploits the symbiotic relationship between edge-cloud computing and LLMs from two key aspects: edge-cloud-empowered LLMs (*Edge4LLMs*) and LLMs-driven edge-cloud computing (*LLMs4Edge*). LLMs have powerful decision-making and personalized service capabilities due to their advanced inference capabilities and rich world knowledge. Meanwhile, edge-cloud computing networks can achieve nearby data collection and processing (e.g., model training and inference) closer to data source. In light of this, we contemplate the integration of edge-cloud computing and LLMs, seeking to harness their synergistic potential. The main contributions are stated as follows:

- First, we provide a comprehensive overview of the characteristics, development and life-cycle of LLMs. Then, we introduce the paradigms of edge-cloud computing and edge AI hardware for the training and inference of LLMs.
- Second, we focus on exploring the symbiotic relationship between edge-cloud computing and LLMs (i.e., *LLMs4Edge* and *Edge4LLMs*), highlighting the bidirectional empowerment.
- Third, we delve into practical applications of LLMs in edge-cloud networks and showcase the integration of LLMs in the realm of different edge-cloud scenes.
- Finally, we outline some research challenges and future directions at the intersection of edge-cloud computing and LLMs integration. Meanwhile, we point out the key areas that should be further explored to improve the performance of LLMs at edge-cloud networks.

The subsequent sections of this survey are meticulously organized as follows: In Section II, we undertake a thorough exploration of LLMs, while Section III is devoted to delving into the foundational aspects of edge-cloud computing networks. The bidirectional synergy between edge-cloud computing and LLMs is unveiled in Section IV (*LLMs4Edge*) and Section V (*Edge4LLMs*). Different practical applications of LLMs within the context of edge-cloud networks are presented in Section VI. In Section VII, we discuss the research challenges and future directions from three different aspects. To bring our exploration to a comprehensive closure, Section VIII encapsulates the concluding remarks of this survey. The organization graph of this survey is shown in Fig. 4.

II. Fundamentals of LLMs

A. Characteristics of LLMs

LLMs stand as a pivotal concept in the realm of DL, which refers to models with an extensive number of parameters and complex structures. These models are often trained with advanced DL techniques and can process large-scale data and learn more complex patterns. In this subsection, we will exploit the key characteristics of LLMs.

One of the key characteristics of LLMs is their large number of parameters [48]. These parameters include model weights and biases. These huge number of parameters are used to capture various complex patterns and intrinsic relationships in input data. Large-scale parameters enable LLMs to better adapt to training data, learn more abstract and complex representations, and improve the generalization ability on unseen data [57]. According to [58], as the model parameters reach a certain quantity, "intelligence" begins to emerge. Typical LLMs may have tens to hundreds of billions of parameters, such as GPT-4, LLAMA2-70B[1] and GLM-130B[2], which depend on their architecture and application domain. However, the huge number of model parameters also requires more computing power resources for training and inference, which can lead to longer training times and higher hardware requirements.

Another notable feature of LLMs is their complex model structure. The complex structure helps the model extract and combine the features of input data layer by layer to form a hierarchical representation, thereby better capturing the deep relationships of input data [50]. The structure of LLMs may also involve some advanced techniques, such as attention mechanisms and residual connections, to enhance model's attention on input data and improve its learning ability. This complexity not only increases the expressive capabilities, but also poses challenges to model training and optimization. Therefore, the training of LLMs usually requires the use of advanced training strategies and technologies. Many public LLMs have been released on HuggingFace[3] platform or embedded into search engines and office software (e.g., *New Bing* and *Microsoft* 365), the readers can easily obtain them.

B. Development of LLMs

In this subsection, we will exploit the evolutionary stages of LLMs from the perspective of parameter scale, technical architecture, modal support and application fields.

From the perspective of parameter scale, LLMs have gone through three stages: pre-training model, large-scale pre-training model, and ultra-large-scale pre-training model. The parameter scale increases by nearly 10 times every year, and the number of parameters has grown dramatically from tens

[1]https://huggingface.co/meta-llama/Llama-2-70b-chat-hf
[2]https://github.com/THUDM/GLM-130B
[3]https://huggingface.co

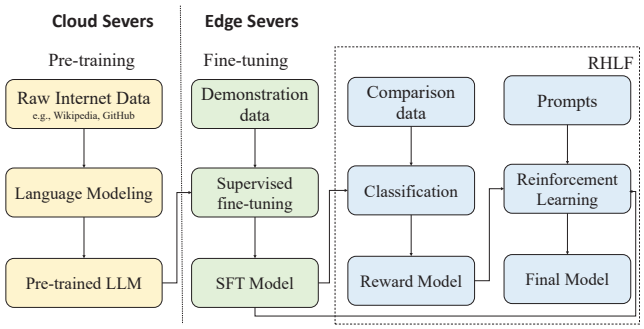


Fig. 5. Overview of LLMops at edge-cloud networks.

of billions to hundreds/thousands of billions [50]. At present, LLMs with hundreds/thousands of billions of parameters have become mainstream, such as OpenAi GPT-4/Sora, Google PaLM [59], Tencent HunYuan-NLP 1T [60] and Alibaba Tongyi Qianwen [61], etc. Their huge amount of parameters allows LLMs to understand and generate content smoothly. These state-of-the-art LLMs usher in a new era of information processing, enabling more nuanced, context-aware interactions between machines and humans.

From the perspective of technical architecture, transformer architecture [62] is the mainstream architectural foundation in the current LLMs, thus forming two main technical routes: BERT and GPT. BERT is knowledgeable in capturing bidirectional contextual information and is particularly suitable for tasks such as question answering, sentiment analysis, and named entity recognition. After the release of GPT-3.5, GPT gradually became the mainstream route for LLMs. GPT's notable advantage over BERT lies in its ability to generate coherent and contextually rich language, enabling more dynamic and versatile language understanding and production [63]. Currently, most LLMs with parameter scales exceeding 100 billion adopt the GPT mode.

From the perspective of modal support, LLMs can be divided into NLP LLMs, CV LLMs and scientific computing LLMs typically [64]. As LLMs continue to evolve, the trend is shifting towards enhancing cross-modal understanding (i.e., multi-modal LLMs). The modalities supported by LLMs are more diverse, from supporting a single task in a single modality of text, image, or video to supporting multiple tasks in multiple modalities. These multi-modal LLMs [65], [66] provide comprehensive analytics and insights for applications such as multi-media processing, decision-making, and more.

From the perspective of application fields, LLMs can be divided into two types: general-purpose LLMs and industry LLMs. General-purpose LLMs have strong generalization capabilities and can complete multi-scenario tasks without fine-tuning or with a small amount of fine-tuning [67]. They are equivalent to AI completing "comprehensive education". ChatGPT, Huawei Pangu [68] and Baidu Wenxinyiyan are both general-purpose LLMs with rich knowledge. The industry LLMs use industry knowledge to fine-tune LLMs, allowing LLMs to complete "professional education" to meet the needs of different fields such as medical [69], finance [70], manufacturing [71], multi-media, etc.

C. Life-cycle of LLMs

In the life cycle of LLMs, the operational processes (i.e., LLMops) unfold through several critical stages:

Tokenization, Embedding and Vector database: The first step in the LLMops usually involves converting raw text into the format that the LLMs can process. Tokenization decomposes text into sequences of words, and then maps these word sequences into high-dimensional vectors through embedding. This embedding operation will capture the underlying structure and relationships in the data, where semantically similar words are mapped close to each other in vector space. To speed up retrieval and similarity comparison, the embedded text vectors are usually stored in vector databases. Such vector databases can efficiently retrieve large-scale text vectors, making tasks such as similarity matching more efficient.

Pre-training and Supervised Fine-Tuning: Pre-training is a foundational phase in the life-cycle of training LLMs [50]. During this stage, the model undergoes extensive training on vast corpora, typically leveraging large-scale text datasets like Wikipedia, GitHub, or some multi-modal databases. The objective is to enable LLMs with a broad understanding of language patterns and semantics. Following pre-training, the pre-trained model as a foundation model (FM) needs to be fine-tuned [72]. This stage trains the model on high-quality demonstration data. The model learns to predict the next word in a sentence and generate the context. The goal is to adapt the pre-trained model to a specific domain, thereby improving the performance of target tasks.

Reinforcement Learning from Human Feedback (RLHF): RLHF serves as a crucial mechanism for refining the output of LLMs based on human-generated evaluations [73]. The fine-tuned model uses DRL to improve its output iteratively. The reward model is built on human evaluation, guiding the model to generate more contextually relevant and human-like language. The process involves a continuous cycle of model generation, manual evaluation, and model refinement. Chain-of-thought prompts are the instructions or questions that the user inputs to LLMs. This may involve designing appropriate natural language queries or other forms of input to guide the model to produce the desired results. Constructing effective prompts can significantly affect the output of LLMs. After the above process, we will get the final model.

Model deployment and inference: This deployment integrates the trained LLMs into real-world environments, which may include cloud platforms, edge devices or other distributed systems. Model inference uses a trained model on real data to generate content or make predictions. This step requires optimized model structures and efficient computing resources to ensure the inference performance. An overview of LLMops at edge-cloud networks can be seen in Fig. 5.

III. Fundamentals of Edge-cloud Computing

A. Paradigm of Edge-Cloud Computing

Typical, edge-cloud computing can be categorized into three distinctive collaborative paradigms: End-edge, edge-edge, edge-cloud, and end-edge-cloud collaboration.

End-edge collaboration. End-edge collaboration [74], [75] represents a transformative paradigm where 'end' refers to devices like smartphones and IoT devices, serving as the front line for data collection and initial processing [76]. These end devices act as the first point of interaction, collecting real-time data through sensing and performing preliminary processing tasks [77], [78]. The 'edge' serves as the hub for model development, fostering continuous refinement and improvement. Typical 'edge' includes the cloud edge, device edge (e.g., NVIDIA Jetson modules, ARM Mbed and Raspberry Pi) and sensor edge (e.g., cameras). Simultaneously, the 'edge' emerges as the execution ground, where models flexibly adapt to real-time scenarios. This decentralized deployment ensures low-latency responses, making critical decisions without the need for constant connectivity to the central cloud servers.

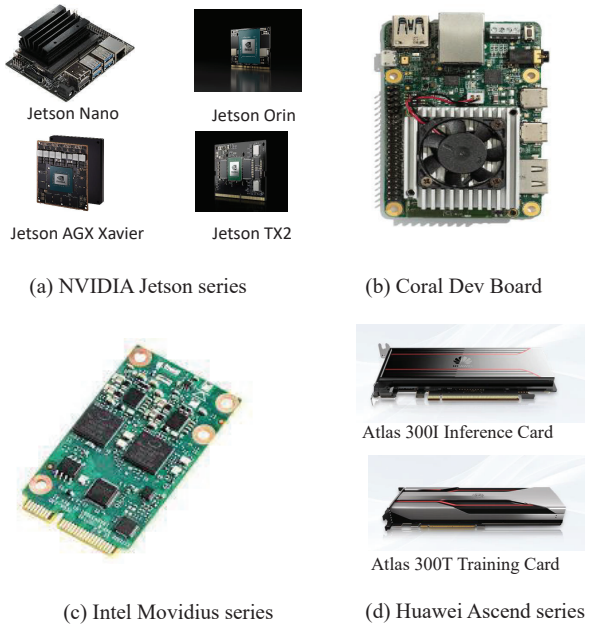
Edge-edge collaboration. Edge-edge collaboration [79] emphasizes collaboration between different edge servers, avoiding the bottleneck of transmitting all computing tasks to cloud servers. In the edge-edge collaborative computing paradigm, computing tasks are distributed to different edge servers at the edge of networks [80]. For LLMs, this means that part of model computation tasks can be distributed to nearby edge servers for training or inference, thereby reducing the burden on cloud servers and overall response delay.

Edge-cloud collaboration. Edge-cloud collaborative computing paradigm [81], [82] combines the advantages of edge computing and cloud computing. In this paradigm, the 'cloud' acts as the intellectual powerhouse, orchestrating the training and evolution of LLMs with vast datasets. Some computation-intensive tasks are assigned to cloud servers for execution, while others are sent to edge servers for processing [83]. For LLMs, part of GPU-friendly inference tasks can be offloaded to cloud servers for processing, especially those complex model structures that require large amounts of computing resources. This paradigm allows LLMs to be efficiently trained and inference in the cloud servers while retaining the low latency and real-time nature of edge processing [84], improving overall system flexibility and performance.

End-Edge-Cloud Collaboration. End-edge-cloud collaborative computing [52], [85], [86] is a comprehensive paradigm that integrates end devices, edge devices, and cloud computing resources. This paradigm allows seamless inter-working between end devices, edge servers, and cloud servers to dynamically allocate computing load based on the nature and requirements of tasks [20], [87]. End devices can perform simple tasks, edge devices can handle tasks of medium complexity, and the cloud is used to handle large-scale, high-computing-demand tasks. For LLMs, different model parts can work together on end devices, edges, and clouds to form a full-stack collaboration computing system. This collaborative paradigm balances flexibility and efficiency in LLM training/inference.

B. Edge AI Hardware for LLMs

The emergence of LLMs like GPT-4 and Sora has indeed brought about a paradigm shift. Nevertheless, it is important to acknowledge that these LLMs have posed significant computational and storage challenges. Innovative advancements in various categories of AI hardware, ranging from processing units to storage chips, including specialized hardware accelerators and architectural optimizations, have played a pivotal role in enhancing training/inference efficiency, and practical application of LLMs.



(a) NVIDIA Jetson series (b) Coral Dev Board

(c) Intel Movidius series (d) Huawei Ascend series

Fig. 6. Different edge AI hardware of NVIDIA Jetson series, Google Coral series, Intel Movidius series, Huawei Ascend series.

1) Computing-Power Processing Units for LLMs: typical processing units can be divided into four categories according to different functions. i) GPU: the highly parallel architecture of GPU provides formidable computational support for the training/inference of LLMs. Its parallel processing capabilities accelerate critical tasks such as matrix computations and facilitate rapid model iteration. Typical products include NVIDIA A100, H100 Tensor Core GPU, AMD's Radeon Instinct series and HUAWEI Ascend Atlas 800 [88]. ii) TPU (Tensor Processing Unit): TPU [89] is specifically optimized for tensor operations. Its deployment in environments such as the Google Cloud platform provides efficient hardware support for LLMs. Typical products include Google TPU-v5e. iii) ASIC (Application-Specific Integrated Circuit): ASIC chips tailored for specific tasks exhibit outstanding performance and energy efficiency, particularly crucial for handling the extensive parameters of language models [90]. Typical products include Intel Gaudi 2 and SAMSUNG Warboy. iv) FPGA (field-programmable gate array). FPGA [91] provides a flexible and programmable hardware acceleration solution, enhancing the efficiency of processing large-scale models through parallel computing and customizable architecture. Typical products include Xilinx series [92], Intel Arria and Stratix.

2) Integrated AI Hardware of LLMs for Edge Nodes: The increasing demands of LLMs underscore the imperative for efficient deployment of AI models at edge nodes. Several prominent solutions for LLMs with edge computing include: i) NVIDIA Jetson series, featuring high-performance GPUs

for both inference and training in edge devices such as drones and smart cameras. ii) Google Coral series, comprising the Edge TPU accelerator and Coral Dev Board, offers efficient edge AI computing solutions, particularly suitable for real-time object detection and speech recognition in edge devices and embedded systems. iii) Intel Movidius series, specialized in visual processing, presents low-power, high-efficiency edge computing solutions applicable to real-time image processing in cameras and surveillance systems, supporting various DL models. iv) Qualcomm AI Engine, integrated into Snapdragon processors, provides high-performance AI computing capabilities suitable for mobile devices and intelligent cameras. v) Huawei Ascend series, including Atlas 300T Training Card, Atlas 300I Inference Card and NPU chips, is designed for efficient AI computation, finding applications in edge scenarios like smart cities and industrial automation, and delivering rapid processing capabilities. As shown in Fig. 6, these mainstream solutions contribute significantly to advancing edge computing and embedding AI applications into the IoT domain.

C. Serving LLMs at Edge-Cloud Networks

In this section, we will discuss some feasible techniques for running LLMs at edge networks.

1) Model Optimization: Deploying LLMs at edge-cloud networks demands a meticulous focus on model optimization to address the challenges of limited computational resources [93], [94]. Optimal model architecture selection is paramount, tailoring it to the specific task while considering the computational constraints of edge devices. Techniques like model pruning [95], [96] and quantization [97] become instrumental in achieving these objectives, allowing for the removal of redundant connections and tolerable loss of accuracy without compromising overall model performance. Moreover, embracing deep compression methodologies, such as model distillation [98], becomes imperative for reducing the model size. This compression aids in minimizing the storage requirements and contributes to more efficient model loading and execution on resource-constrained edge devices. The intricate balance between model complexity and efficiency is a key consideration in achieving optimal performance in edge environments.

2) Edge Device Compatibility: Ensuring compatibility with edge devices is another critical factor for deploying LLMs at edges. This necessitates a meticulous approach to align the model with the diverse hardware configurations prevalent in edge environments. Firstly, it involves adapting LLMs to the specific hardware architecture of edge devices. This may require optimizations and fine-tuning to leverage the capabilities of different processors and accelerators. Besides, compatibility checks for runtime libraries and dependencies are paramount to guarantee a seamless execution environment, preventing conflicts and ensuring the model's smooth operation on diverse edge devices. Meanwhile, LLMs should execute within the limited computational power and also be optimized to minimize energy consumption. It is necessary to select algorithms and optimization strategies that align with the low-power characteristics of edge devices to strike a balance between model performance and resource efficiency.

3) Containerization: Containerization technique provides a versatile and standardized environment for seamless execution across diverse edge devices [99]. LLMs along with their runtime environment and dependencies, can be encapsulated in lightweight docker containers. This encapsulation facilitates easy portability, ensuring that the model and its associated components can be seamlessly transferred across different edge devices, regardless of their underlying infrastructure. Apart from this, container orchestration tools, such as Kubernetes [100], [101], allow for the efficient coordination of containerized language models, ensuring their availability, scalability, and reliability across edge and cloud environments. Administrators can easily manage resources, automate scaling based on demand, and streamline updates without disrupting ongoing operations. Each LLM instance resource isolation and runs within its dedicated container, preventing interference or conflicts with other instances. This enhances both security and reliability, as issues with one instance are contained within its container, safeguarding the overall system stability.

4) Effective Orchestration: Effective orchestration between edge and cloud servers is crucial for deploying seamlessly, ensuring optimal performance and resource utilization. Kubernetes is a container orchestration tool that provides a unified framework and enables consistent deployment practices, ensuring that LLMs can be deployed with ease irrespective of the underlying architecture. Besides, load balancing can be facilitated through container orchestration tools. In the context of LLMs, this ensures that requests are evenly distributed across diverse edge and cloud nodes, preventing the overloading of specific resources and optimizing response delay. In addition, version management is another critical aspect. Orchestrating the deployment of LLMs involves handling multiple versions coherently. This enables the seamless rollout of new model versions while retaining the ability to revert to older versions if necessary, ensuring a smooth and controlled transition. In short, effective orchestration streamlines the deployment, scaling, and management of LLMs. It incorporates automatic scaling, version management, and load balancing to ensure efficient and reliable operation across a distributed network of edge and cloud resources.

IV. Edge-cloud Computing Empowered LLMs

In this subsection, we discuss how LLMs empower edge-cloud computing (i.e., *Edge4LLMs*) from the aspects of data, computing power, model training and inference.

A. Edge Data Management for LLMs

Training LLMs requires large-scale, high-quality, multi-modal data [102]. These multi-modal data can be in various forms (e.g., text, image, audio or video, etc.) that enable LLMs to understand the world more comprehensively and thus better solve problems such as image description generation or text understanding. Multi-modal data provide extensive information and enhance the generalization ability required by LLMs. In this subsection, we explain how edge-cloud computing can empower LLMs from the perspectives of data

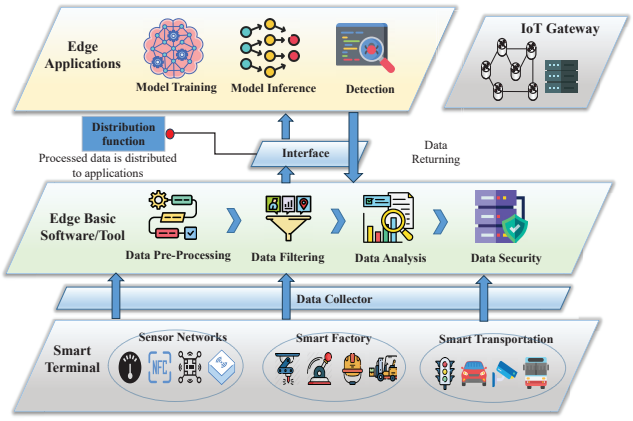


Fig. 7. Paradigm of edge data management for LLMs.

collection, data pre-processing and data security. Fig. 7 shows the paradigm of data management for LLM at network edges.

1) Multi-Modal Data Collection: edge servers are generally closer to data source and can achieve a lower data transmission cost [103], [104]. Therefore, it is necessary to collect multi-modal data from various fields and multiple data sources with edge servers. This can significantly reduce the expense of data collection and make it easier to collect more data. One feasible approach is a network-based data collection strategy. Network-based data collection strategies collect multi-modal data for supporting training and inference of LLMs based on edge network infrastructure. By harnessing the power of the edge network, edge servers can seamlessly gather diverse data types, including text, images, audio, and video, enabling LLMs to acquire the necessary knowledge and generate contextual responses. Zheng *et al.* [105] design a novel framework for graph data collection in distributed IoT networks and propose centralized and two decentralized algorithms that aim to minimize bandwidth consumption for addressing the challenges of overlapping and sensitive data, respectively. The authors in [106] propose a distributed data collection framework, named *DaaC*. *DaaC* emphasizes privacy and efficiency, and allows for secure and efficient data gathering in complex multi-level edge networks. Apart from this, Liu *et al.* [107] present a collaborative data collection approach in Internet-of-Vehicles (IoV) to achieve better spatio-temporal data evenness, addressing the limitations of vehicle data collection in vehicular edge computing (VEC) networks.

These diverse contributions emphasize the importance of edge computing for efficient data collection, privacy preservation, and collaborative strategies to address evolving challenges. Based on the above studies, we summarize the potential key procedures of network-based data collection strategy as: i) network data parsing: edge servers parse and process network data streams (e.g., real-time streams, cameras or sensor data, etc.). ii) multi-modal data fusion: edge servers fuse multi-modal data from different sensors or cameras to generate richer content. For example, image data captured by cameras can be combined with audio data from sound sensors to provide a more comprehensive scene understanding. iii) multi-modal data transmission: edge servers use data compression and optimization techniques to transmit collected multi-modal data to cloud servers for further use while maintaining high data quality. This efficiently facilitates the transmission of multi-modal data under limited conditions of network edges.

2) High-efficiency Data Pre-Processing: Data pre-processing is crucial to improve model robustness and reduce resource costs of training and inference of LLMs. Cleaned high-quality data is the basis for building efficient, accurate and reliable LLMs [48], [108]. In the first step of data pre-processing, it is necessary to check whether the data format is as expected (including data deduplication, filtering and data missing value processing, etc,.). During the cleaning process, the data needs to be standardized and normalized to ensure that the data has consistent metrics for subsequent analysis and model training [109]. Edge servers can verify the integrity of the data, whether the data type is correct, and whether the data contains any outliers [110]. This paper [109] introduces a two-part framework for IoT data pre-processing at edge networks. The first part comprises a design tool for simplifying the AI data pre-processing phase through generalization and normalization. The second part is an IoT tool adopting edge-cloud collaboration to enhance data quality progressively. To address big data cleaning challenges in industrial Sensor-Cloud Systems (SCSs), Wang *et al.* [110] propose data cleaning methods that utilize edge nodes during data collection, incorporating an angle-based outlier detection method and support vector machine (SVM) for improving data cleaning efficiency, and reducing bandwidth and energy consumption in the industrial SCSs.

3) Reliable Data Security: Edge-cloud computing architectures inherently facilitate decentralized processing and computation on edge devices, diminishing the necessity for transmitting data to cloud servers [111]. This localization process can minimize the risk of raw data during transmission, thereby enhancing data privacy protection. For LLMs, one feasible approach is to use FL, where only the model parameters are transmitted rather than raw data [112]. Some studies first carried out research on integrating the foundation models (FMs) of LLMs and FL [113]. This paper exploits the symbiotic relationship between FMs and FL. FL enhances FM through collaborative training, while FMs provide a robust starting point for FL, accelerating convergence and improving training performance. To achieve privacy-preserving and collaborative learning across multiple end-users, Yu *et al.* [114] combine the strengths of FMs like LLaMA, BERT and GPT with FL. The proposed approach addresses privacy concerns associated with optimizing FMs, enabling collaborative learning without compromising sensitive data. Besides, the authors also outline the benefits and challenges of integrating FL throughout the FM lifespan, including pre-training, fine-tuning, and application. These studies [113], [114] have proven that integrating LLMs and FL can enhance training efficiency and ensure data security to a certain extent.

In addition, some authors also study how to protect the data security of FL for LLMs from other aspects. Kuang *et al.* addresses the challenges of privacy-preserving fine-tuning of LLMs in resource-constrained edge environments. The proposed package, *FS-LLM* [115], offers an end-to-end pipeline

TABLE III
SUMMARY OF DATA COLLECTION, PRE-PROCESSING AND SECURITY OF EDGE DATA MANAGEMENT AT EDGE-CLOUD NETWORKS.

	Ref.	Scenarios	Objective	Benefits/Main Ideas	Tools /Method
Data Collection	[105]	Distributed IoT networks	Minimize the bandwidth consumption	Address the challenges of overlapping and sensitive data for graph data collection	Centralized and decentralized algorithms
	[107]	IoV networks	Achieve better spatio-temporal data evenness	Address the limitations of vehicle data collection in IoV networks	Online and Offline Scheduling
Data Pre-Processing	[109]	Edge networks	Improve data pre-processing efficiency	Solve the IoT data pre-processing at network edges	Two-part framework
	[110]	Industrial SCSs	Enhance big data cleaning efficiency while reducing bandwidth and energy consumption	Utilize edge server and incorporating an angle-based outlier detection method and SVM for improving big data cleaning efficiency	SVM algorithm
Data Security	[113]	\	Improve the performance of FMs with FL	Exploit the symbiotic relationship between FMs and FL, while FL enhances FM through collaborative training and FMs provide a robust starting point for FL	Federated FMs
	[114]	Edge networks	Achieve the privacy-preserving and collaborative learning across multiple end-users	Address the privacy concerns associated with optimizing FMs	Federated Pre-training and Fine-tuning

automating dataset that integrates pre-processing, federated fine-tuning, and performance evaluation of LLMs. To tackle the challenges of limited labelled data and communication constraints in mobile FL scenarios, the authors propose a low-parameter FL (*LP-FL* [116]) approach that combines few-shot prompt learning for training of LLMs. *LP-FL* leverages the low-rank adaptation (LoRA) technique to reduce computation and communication costs, which outperforms full-parameter FL in sentiment analysis tasks across various FL settings. In terms of the FL framework for LLMs, *FATE-LLM* [117] promotes efficient training through parameter-efficient fine-tuning methods while safeguarding intellectual property and preserving data privacy during both the training and inference process. These studies show that FL can guarantee data privacy throughout the entire lifecycle of LLM. We summarize the strategy of data collection, pre-processing and security of edge data management in Table III.

B. Edge Computing Power for LLMs

With the rapid growth of IoT devices and applications, there are a large number of scattered and idle computing power resources at network edges. LLM training and inference often require large amounts of computing power resources. Edge-cloud networks can effectively integrate these computing power resources for the training and inference of LLMs. In this subsection, we will discuss the edge computing power for LLM from the aspect of computing power supply, scheduling and trading, respectively.

1) EC for Optimizing Computing Power Supply: Edge plays a pivotal role in computing power supply by strategically distributing computational tasks and efficiently utilizing available resources [118], [119]. To achieve it, computing power pooling is a feasible resource management strategy designed for aggregating and sharing these idle computing resources (such as CPU, GPU, TPU, etc,.), so that edge devices or DL applications can jointly access and utilize these computing power more efficiently. This technology can play a key role in the training and inference of LLMs at edge networks. Tang *et al.* propose a computing power supply framework *FusionAI* [119] that harnesses the computational power of consumer-level GPUs for training and deploying LLMs. *FusionAI* use a broker with backup pool to dynamically join and quit of computing power providers. Meanwhile, *FusionAI* shows that 50$\times$ RTX 3080 GPUs can match the throughput of 4 more expensive H100 GPUs. This makes decentralized LLM training and deployment more accessible and cost-effective.

In a related domain, the authors in [120] design *In-Network Pooling* in computing power networks (CPN). They utilize a modified DRL framework to efficiently utilize idle computing and caching resources for resource-hungry applications. *ERP* [121] is a typical edge resource pooling framework that exploits computation resource collaboration and sharing in mobile computing networks. Additionally, Lyu *et al.* [122] propose the framework *SoSA*, a novel architecture in large-scale WiFi systems to create a resource-pooled edge system. To address the challenges of low resource utilization, *SoSA* can share the pooled resource conveniently and improve service performance obviously. Together, these innovation frameworks contribute to the evolution of computing power supply strategies in distributed edge environments.

2) EC for Optimizing Computing Power Scheduling: edge computing can optimize computing power scheduling by bringing computation resources to where they are needed, thereby efficiently utilizing these idle resources [123]. This proximity to data sources allows for real-time processing and enhances overall system performance. We list some typical strategies for optimizing computing power scheduling at edge networks: resource-aware computing power scheduling, dynamic computing resource adjustment and intelligent computing resource allocation.

Resource-aware Computing Power Scheduling: resource-

aware computing power scheduling is a resource management strategy in edge environments. The system will sense and monitor the resource status on edge servers to intelligently allocate computing power to meet task requirements [15], [124]. We have listed some innovative resource-aware computing power scheduling frameworks. Xu *et al.* [15] proposes *COSREL*, a co-scheduling framework for efficient concurrent inference of DNN models on heterogeneous mobile and edge devices. *COSREL* utilize heterogeneous computational resources (e.g., CPU, GPU, etc), dynamically adjust scheduling decisions based on system runtime, and strike a balance between latency, throughput, and energy efficiency, while maintaining the model accuracy. This resource-aware scheduling strategy can improve availability and resource utilization, ensuring ML/DL models run efficiently on edge nodes. The authors in [124] delve into the modelling and optimization of computing power resource awareness in CPNs. The proposed approach introduces a novelty structured data model for computing power resources and designs a dynamic scheduling algorithm to manage resource-aware events.

Dynamic Computing Resource Adjustment: Dynamic resource adjustment strategy allows the edge network to adjust the allocation of computing resources based on actual needs dynamically [125], [126]. This means that if LLMs require more computing resources, the system can respond promptly and reallocate resources to meet the task's requirements. In practical applications, dynamic resource adjustment can be achieved through resource management systems and automated algorithms. The system can continuously monitor the status of tasks and resource utilization, and reallocate resources according to pre-determined strategies. These strategies are highly beneficial for LLMs training and inference, as these tasks often require large amounts of computing resources and the task load may change over time. Murakami *et al.* [127] design a priority-aware guaranteed hardware resource allocation approach to improve resource utilization and service capacity. The proposed approach addresses inefficiencies in previous hardware allocation by enabling dynamic reallocation based on service demand changes. Apart from this, the authors in [128] propose a dynamic resource orchestration model that dynamically adjusts the computing resources assigned to given jobs and addresses the challenge of efficiently processing data from edge devices in the cloud-to-edge continuum.

Intelligent Computing Resource Allocation: edge networks can use intelligent algorithms (e.g, DRL) to allocate computing resources based on specific task requirements [129], resource prediction [130] or other performance indicators. This ensures that tasks are assigned to the appropriate edge nodes to minimize latency and improve responsiveness. Li *et al.* [129] exploits the application of asynchronous DRL in VEC for collaborative task computing and on-demand resource allocation with required delay strict. The collaborative computing framework can facilitate intelligent management of diverse resources across vehicles, edge servers, and the cloud. Chien *et al.* in [130] presents a dynamic resource prediction and allocation solution with Edge AI for cloud radio access networks. Specifically, they utilize long short-term memory for dynamic throughput prediction and a genetic algorithm-based approach for optimized resource allocation. Additionally, the authors in [131] propose a graph-based model for dynamic resource allocation and address the need for unified resource management in the cloud-to-edge continuum.

3) EC for Optimizing Computing Power Trading: Edge computing is emerging as a crucial enabler for the optimization of computing power trading. By trading idle computing resources at network edges, it can flexibly respond to fluctuations in task demands, thus achieving greater flexibility and cost-efficiency [132]. A typical architecture design of computing power trading in edge-cloud computing networks can be shown in Fig. 8. Intelligent computing power trading mechanism (e.g., auction/incentive mechanism) empowers stakeholders in edge-cloud computing ecosystems, such as cloud service providers (SPs), edge infrastructure operators and mobile users, to engage more granular [133]. The following are several common computing power trading strategies in edge-cloud networks:

Computing Resource Leasing: The LLM owners can rent computing resources on edge devices to perform training and inference of LLMs [134]. Users typically incur charges based on the usage of resources or the duration of usage. This strategy is suitable for users who need flexible use of computing resources. We list some computing resource leasing works at edge-cloud networks as follows. Chen *et al.* [134] propose a trade framework *EETORP* that models the computing resource between edge nodes and users by stackelberg game theory and formulate the problem as a computing resource pricing and purchasing game. Furthermore, Li *et al.* [133] exploit a three-tier edge computing market involving edge servers, brokers, and users. To maximize social welfare, the authors design a pricing-based resource allocation mechanism called *MECM* to allocate edge computing resources and decide prices, facilitating efficient resource allocation in this complex ecosystem for the benefit of all stakeholders.

Task Delegation and Service Level Agreement (SLA) Model: Users can delegate their LLMs training or inference tasks to the owner of edge devices or professional providers. The equipment owner or provider is responsible for performing the tasks and is billed for the number of tasks or work done. In the SLA model, a contract is signed between the user and computing resource owner or SPs that stipulates the conditions [135], quality, and fees for LLM task execution. Du *et al.* [136] exploit a blockchain (BC) aided edge market where data service operators rent computing resources to smart terminals, ensuring trustworthiness through a smart contract-based trading mechanism. The authors in [137] design a hierarchical BC-based computing resource trading scheme for multi-access edge network slicing. They address the challenge of efficient resource management and trading among mobile virtual network operators by proposing a secure transaction framework using hyperledger smart contracts.

Computing Resource Trading Platforms: There are some resource trading platforms that enable the trading of computing resources between users and computing resource owners. These trading platforms (such as Amazon *AWS*, *EdgeMicro*, *Edgevana*, etc) can provide market or auction mechanisms, where users can perform tasks on different devices according to their needs, and device owners can provide resources and

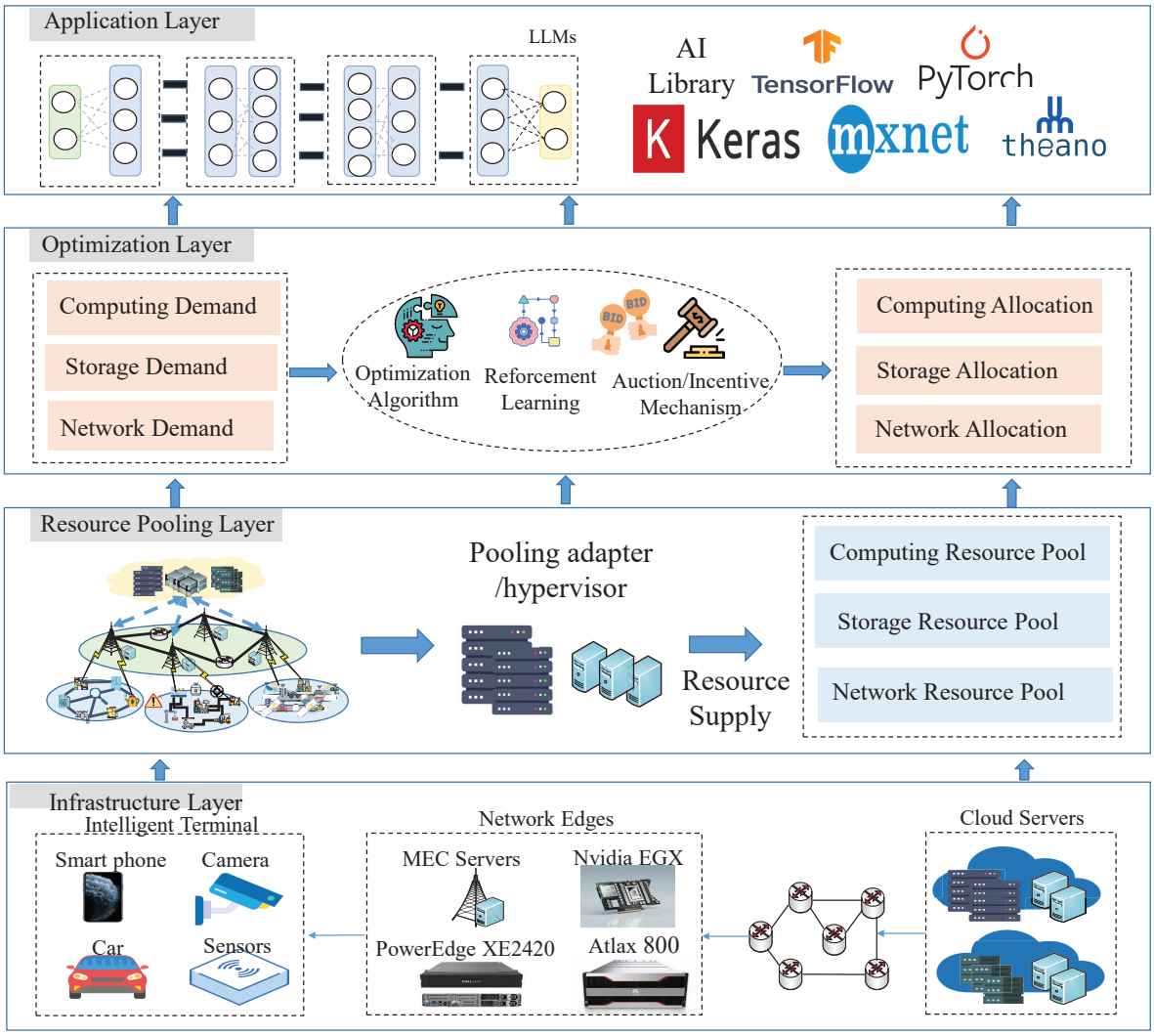


Fig. 8. Architecture design of computing power trading in edge-cloud networks.

receive compensation. *PPIO* edge cloud [138] is a powerful computing power trading platform that provides users with efficient and customizable computing power resource management solutions through diverse computing resource matrices and flexible product selection. The authors propose a BC-based cloud-edge computing power trading framework *AI-Bazaar* [139]. *AI-Bazaar* is designed to address challenges such as low utilization of computing power and inefficiency in AI services management. AI consumers can rent computing power from providers for BC mining and AI services. These computing power trading platforms have proven that computing power utilization can be significantly improved through trading. The strategy, scenarios, objective, benefits/challenges, and tools/methods for edge computing power optimization are summarized in Table IV.

C. Edge for LLM Training

Edge computing accelerates LLM training through three key mechanisms: parallel training, leveraging parallel processing capabilities; distributed collaboration, facilitating cooperative efforts across diverse devices; and incremental learning, enabling continual model updates. This holistic approach enhances the efficiency and scalability of LLM training in edge computing environments.

1) Parallel Training: Parallel training at edge computing networks is mainly performed in the following ways: data parallelism, model parallelism and pipeline parallelism.

Data Parallelism: Data parallelism [140], [141] allows training data distributed across multiple edge devices. Each device is responsible for processing part of the training data to speed up the training process of LLMs. In the early days, data parallelism was applied to deep neural networks (DNNs). Liu *et al.* propose a hierarchical edge AI learning framework *HierTrain* [140] for efficient DNN training in mobile-edge-cloud computing environments. *HierTrain* utilizes a novel hybrid parallelism method to adaptively distribute DNN layers and data samples across edge devices, edge servers, and cloud servers. By optimizing scheduling at both layer and sample granularity, *HierTrain* significantly reduces training time and achieves up to 6.9× speedup.

Furthermore, Li *et al.* in [141] focuses on optimizing distributed model training in edge computing-enabled optical

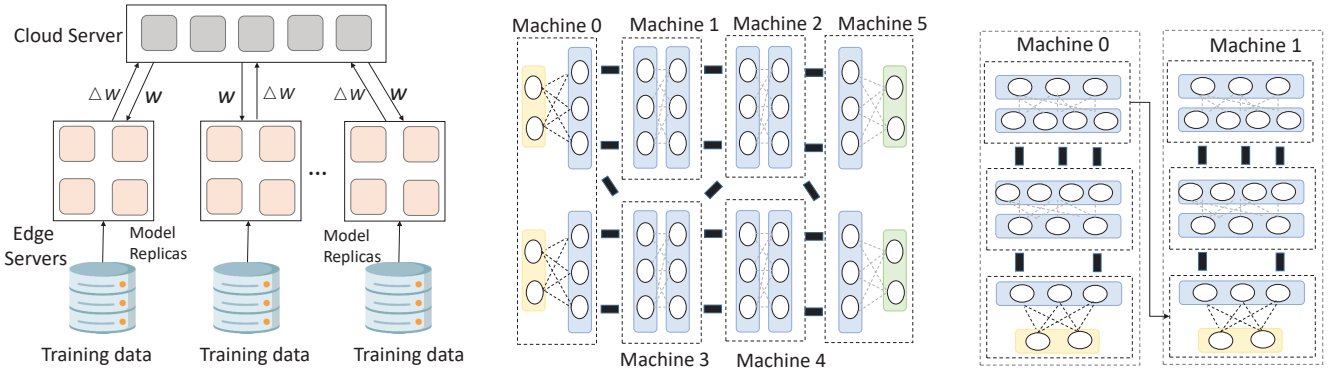
TABLE IV
SUMMARY OF STRATEGY, SCENARIOS, OBJECTIVE, BENEFITS/CHALLENGES, AND TOOLS/METHOD IN TERMS OF EDGE COMPUTING POWER FOR LLMS.

Ref.	Strategy	Scenarios	Objective	Benefits/Main Ideas	Tools /Method
FusionAI [119]	Optimizing computing power supply	Massive consumer-Level GPUs environment	Improve the computing power supply efficiency	Harnesses the computational power of consumer-level GPUs for training and deploying LLMs.	System design
In-Network Pooling [120]	Optimizing Computing Power Supply	CPNs	Maximize the long-time system utility	Efficiently utilize idle computing and caching resources for resource-hungry applications.	DRL
[128]	Dynamic Computing Resource Adjustment	Cloud-to-edge continuum	Improve the resource orchestration efficiency	Propose a dynamic resource orchestration model that dynamically adjusts the computing resources assigned to given jobs	Dynamic resource allocation
[130]	Intelligent Computing Resource Allocation	Cloud radio access networks	Optimize resource allocation	Utilize long short-term memory for dynamic throughput prediction and optimized resource allocation	Genetic algorithm
MECM [133]	Computing Resource Leasing	Three-tier edge computing	Maximize social welfare	Allocate computing resources and decide prices for the benefit of all stakeholders	Pricing-based resource allocation mechanism
EETORP [134]	Computing Resource Leasing	MEC	Minimize the total costs of devices and maximize the benefits of ESs	Design a trade framework that models the computing resource between edge nodes and users	Stackelberg game theory
AI-Bazaar [139]	Computing Resource Trading	Cloud-Edge computing network	Maximize profit	Address the challenges of computing power inefficiency in AI services management	Multi-agent reinforcement learning

networks. They efficiently partition and distribute training data among different edge nodes and propose a data parallelism deployment algorithm DPDA to solve the formulated problem. To apply data parallelism in LLMs, Kuang *et al.* in [115] implement package *FS-LLM* that provides data parallelism with multi-GPU memory optimization and CPU offloading capabilities. The authors in [142] present a synchronous training framework *DAPPLE* that combines data parallelism and pipeline parallelism for LLMs. They introduce a novel parallelization strategy planner to optimize partition and placement, achieving up to 3.23× speedup compared to PipeDream's planner. These innovation frameworks in data parallelism offer not only parallel processing capabilities but also innovative strategies for enhanced speed and efficiency in handling substantial model sizes.

Model Parallelism: Model parallelism [143] allows splitting LLMs into multiple smaller parts and distributing these parts to multiple edge devices for parallel training. This approach can significantly reduce the complexity of LLM training process and the computational burden of each edge device. Sen *et al.* in [143] present a distributed DL system *DMP* that realizes parallel training over edge devices. Specifically, *DMP* optimizes training on edge nodes by clustering them and utilizing both heuristic and DRL-based algorithms to partition DL models to edge nodes for model parallelism. Apart from this, the authors in [144] propose a novel fused-layer-based DNN model parallelism approach with computation offloading by converting DNN layers into smaller ones that enable model inference on IoT devices in MEC networks. To reduce communication costs and improve memory efficiency, the authors in [145] propose an adaptive and resilient model parallelism distributed inference framework, named *AR-MDI*. *AR-MDI* is implemented and tested on NVIDIA Jetson TX2 and can optimally allocate DNN models across edge workers. Li *et al.* in [146] design a novel serving system *AlpaServe* that leverages model parallelism to enable statistical multiplexing for large DL models, which can reduce serving latency for bursty workloads. Therefore, model parallelism proves to be a potential solution for accelerating large-scale model training at edge-cloud networks.

Pipeline Parallelism: Pipeline parallelism [147], [148] is an efficient processing method for LLMs in edge networks. Applying this approach, LLMs can be decomposed into multiple sub-models or processing stages. Each stage is responsible for performing specific tasks and can run in parallel on edge devices. The difference between data parallelism, model parallelism and pipeline parallelism is depicted in Fig. 9. Currently, model parallelism is widely used in the training of DNN and CNN. This paper designs a novel approach *ResPipe* [147] for DNN training at network edges and employs a pipeline parallelism learning method. Experiment on Android-based smartphones demonstrates the proposed approach can improve convergence rates and accuracy for CNN training. Chen *et al.* present a fault-tolerant pipeline-parallel distributed training approach *FTPipeHD* [148] on heterogeneous edge devices at MEC networks. *FTPipeHD* enables distributed training across diverse devices, optimizing partition points based on real-time computing capacities, which can significantly improve model training efficiency. In conclusion, the integration of pipeline parallelism, exemplified by *ResPipe* and *FTPipeHD*, demonstrates a promising avenue for the efficient processing of LLMs in edge networks in the future. We list some related studies on data parallelism, model parallelism and pipeline parallelism in Table V.



(a) Data Parallel. (b) Model Parallel. (c) Pipeline Parallel.

Fig. 9. Comparison of data parallelism, model parallelism and pipeline parallelism.

TABLE V
SUMMARY OF SCENARIOS, MAIN IDEAS AND PERFORMANCE IMPROVEMENTS IN TERMS OF DATA, MODEL, PIPELINE PARALLELISM OF LLMS.

	Name/Ref	Scenarios	Main ideas	Performance improvements
Data Parallelism	HierTrain [140]	Mobile-edge-cloud computing networks	Utilize a novel hybrid parallelism method to adaptively distribute DNN layers and data samples across edge devices, edge servers, and cloud centers	Reduce training time and achieves up to 6.9× speedup
	DPDA [141]	Edge computing -enabled optical networks	Efficiently partition and distribute training data among edge nodes	Reduce by about 10% in terms of blocking rate compared with the benchmark
Model Parallelism	DMP [143]	MEC	Optimize training on edge nodes by clustering them and utilizing DRL-based algorithms to partition DL models	DMP-H and DMP-RL can perform 29-31%, 29-39%, and 24-36% performance improvements in terms of training time
	PSOMW [144]	MEC	Propose a novel fused-layer-based model parallelism approach by converting DNN layers into smaller ones that enable inference on IoT devices	Reduce the DNN inference time by an average of 12.75 times compared to baseline schemes
	AR-MDI [145]	Mobile-edge-cloud computing networks	Optimally allocate DNN models across different edge workers	AR-MDI can reduce the processing time by 80.2% and 44.5% as compared to baseline schemes
Pipeline Parallelism	ResPipe [147]	MEC	Employ pipeline parallelism method to reduce communication and storage costs.	Improve convergence rates and accuracy for CNN training
	FTPipeHD [148]	MEC	Enable distributed training across diverse edge devices, optimizing partition points based on real-time computing capacities	FTPipeHD is 6.8 times faster in training than the state-of-the-art method

2) Distributed Collaboration: There exists large available idle computing resources at network edges. Edge devices can process distributed training [149], [150] and perform local model updates. After that, they transmit the updated parameters to the cloud servers or other devices for global parameter aggregation [53]. It will facilitate cooperative efforts across diverse devices. We list some studies on distributed collaboration at edge-cloud environment [149], [151], [152]. Ren *et al.* [151] exploit the acceleration of distributed DNN training for both CPU and GPU scenarios to improve training speed in wireless federated edge networks. To train ML models on data-distributed edge nodes, Wang *et al.* in [149] propose a control algorithm that optimizes the balance between local updates and global parameter aggregation for minimizing training loss within resource constraints and without transmitting raw data to a central server. The authors in [152] propose a distributing DNN training framework across IoT edge devices to reduce the workload of cloud servers and minimize communication traffic. They adopt a heuristic technique for creating smaller networks on edge devices through knowledge distillation.

Building on the idea of hierarchical FL, Liu *et al.* in [150] introduce an efficient client-edge-cloud hierarchical FL to enable multiple edge servers to perform partial model aggregation, which achieves faster training and communication-computation trade-offs. In summary, these studies collectively contribute to the optimization of ML/DL model training on edge devices. They address the challenges related to resource constraints, communication traffic, and training speed, providing a comprehensive perspective on the evolving landscape of decentralized ML.

3) Incremental Learning: LLMs on edge devices can be continuously improved through incremental learning without re-training. This approach allows the model to be locally updated based on new data [153]–[156], as shown in Fig.

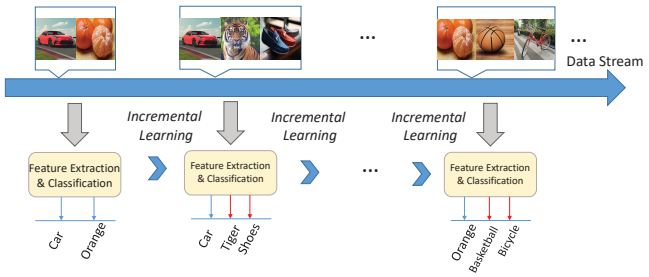


Fig. 10. Mechanism of incremental learning.

10. Gong *et al.* [157] exploit intelligent cooperative edge computing in IoT networks, aiming to seamlessly integrate AI and edge computing. The authors redesign AI-related modules to distribute core AI functions from the cloud to edge, enabling incremental model updates efficiently. The authors in [154] propose a novel incremental learning framework for DNNs, named *LwS* by sharing and cloning fully connected layers and reducing training complexity and memory requirements, enabling efficient incremental learning on edge devices that have idle memory and computation resources.

In addition, some scholars also apply incremental learning to other applications or problems. Bu *et al.* [155] design the IHoPCM algorithm for incremental medical data co-clustering across multiple hospitals in edge-cloud networks. It employs DL models to analyze medical data on edge networks, then performs incremental high-order possibilistic c-means clustering on the cloud. Liu *et al.* [156] propose an incremental clustering-assisted popularity learning algorithm to adapt to changing trends for addressing the content caching problem with unknown and time-varying popularity profiles. In summary, incremental learning at the edge empowers LLMs to dynamically adapt to evolving environments and data distributions. This continual enhancement boosts real-time performance and mitigates the need for extensive bandwidth usage, as only updated model parameters need to be transmitted.

D. Edge for LLM Inference

In this subsection, we will introduce some common optimization methods (e.g., quantization and knowledge distillation) for LLM inference on resource-constrained edge devices.

1) Quantization: model quantization is used to convert the weights and activation values of LLMs into low-bit integers [158], thereby reducing computing and memory requirements and improving inference efficiency on resource-constrained edge devices. This method effectively reduces model size and speeds up inference process. Common used quantitative techniques for LLMs include: post-training quantization (PTQ) [82], [159] and quantization-aware training (QAT) [160].

Post-Training Quantization: PTQ quantifies the trained ML/DL model and converts the model's floating point parameters (weights and activation values) into low-bit integers to reduce the model's memory footprint and compute resource usage [161], [162]. Currently, PTQ is widely used in edge-cloud networks. Huawei Cloud develops *Auto-Split* [159], a framework for collaborative edge-cloud AI by jointly

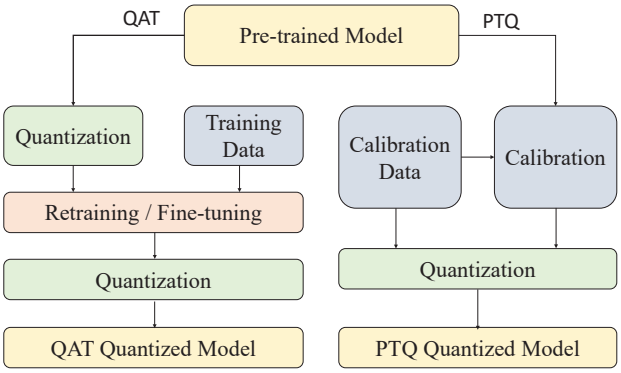


Fig. 11. Comparison of PTQ and QAT.

applying model splitting with PTQ to edge DNN in edge-cloud networks. *Auto-Split* can address the challenge of efficiently deploying large ML/DL models on edge devices while maintaining high accuracy and low latency. Wang *et al.* in [163] propose *QuantPipe*, a communication-efficient distributed edge system that employs adaptive PTQ to compress tensors, responding to dynamic bandwidth changes in edge environments with limited inference accuracy loss.

Many studies investigate the integration of PTQ techniques in LLMs. Kwon *et al.* design *AlphaTuning* [164] that adopts PTQ of the pre-trained language model and uses binary-coding quantization to factorize parameters, freezing binary values while fine-tuning scaling factors during adaptation. The authors in [165] propose *FPTQ*, a novel fine-grained PTQ method for LLMs. *FPTQ* combines the benefits of 4-bit weight quantization and 8-bit matrix computation, addressing performance degradation with layerwise activation quantization strategies. It achieves state-of-the-art quantized performance on standard benchmarks like BLOOM, LLaMA, and LLaMA-2 for deploying LLMs, enabling wider real-world applications without requiring further fine-tuning. Note that precision loss during quantization needs to be carefully handled to ensure that it does not impact the performance of critical applications.

Quantization-Aware Training: Unlike traditional PTQ technology, QAT operates during training and allows the model to consider the impact of quantization while maintaining the accuracy [162], [166]. The comparison of PTQ and QAT is shown in Fig. 11. Several research endeavors have demonstrated the feasibility of conducting QAT on resource-constrained devices. Zhou *et al.* propose a QAT framework *Octo* [160] for efficiently enabling on-device learning by 8-bit fixed-point (INT8) quantization in both forward and backward passes. *Octo* shows promise for enabling tiny on-device learning with improved processing speed and memory reduction on commercial AI chips.

In terms of applying QAT in LLMs, Liu *et al.* in [167] propose a novel framework LLM-QAT. It employs data-free distillation leveraging pre-trained model generations, enabling the quantization of generative models independent of training data. The authors in [168] design *GPUSQ-TLM*, a compression scheme for transformer-based language models (TLMs) that leverages GPU-friendly fine-grained structured sparsity and

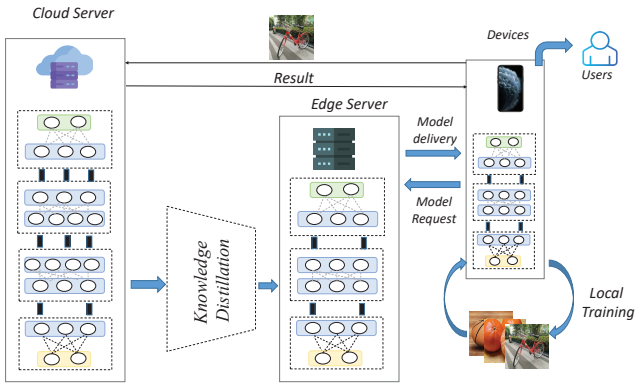


Fig. 12. A feasible knowledge distillation solution at edge-cloud networks.

quantization. It prunes and quantizes dense TLM models by QAT to meet GPU acceleration constraints, achieving state-of-the-art compression with minimal accuracy loss. In summary, QAT may increase the complexity and computational cost of the training process. Therefore, application scenarios and resource constraints must be carefully considered when selecting a suitable quantification method. We summarize the main ideas and performance improvements of related studies on PTQ and QAT in Table VI.

2) Knowledge Distillation (KD): Since edge servers often have limited computing resources, deploying LLMs may face challenges, such as memory consumption. To solve the problems, KD technology can transfer knowledge from a larger model to a smaller model, and then deploy the smaller model on edge devices for inference [169]. Fig. 12 shows a feasible knowledge KD at edge-cloud networks. The teacher model can be a larger model trained in cloud servers or data centres, while the student model is a small model deployed on edge devices. The goal of KD is to enable the student model to capture the knowledge of the teacher model by minimizing the difference between the student model's predictions and the teacher model's predictions [170].

The main KD methods include: pre-training distillation [171], [172], multi-layer distillation [173], [174] and multi-step distillation [175]. Pre-training distillation uses a larger teacher model (e.g., BERT, GPT) that has been pre-trained on a large-scale dataset to guide the training of a relatively smaller model (i.e., student model). This process aims to transfer the knowledge of the teacher model to the student model so that the student model approaches or matches the teacher model in performance while maintaining a smaller model size. The authors in [171] propose *DistilBERT* for efficient on-the-edge and resource-constrained NLP applications. Through KD during pre-training, it achieves a 40% reduction in model size, 60% faster inference speed, and retains 97% of the language understanding capabilities of its larger counterparts. Apart from this, Zhang *et al.* design an innovative task-specific KD framework AD^2 [176] for large pre-trained transformer models like BERT and GPT-3. AD^2 can achieve student models with 99.6% accuracy of teacher model while outperforming task-specific knowledge distillation baselines by 1.2%. He *et al.* in [172] propose *KDEP* as an alternative strategy for efficient pre-training. *KDEP* leverages KD techniques to transfer the learned feature representations from existing pre-trained models to new student models, achieving comparable performance to supervised pre-training.

In the evolving landscape of KD, researchers continue to exploit innovative techniques such as multi-layer distillation [173], [174] and multi-step distillation [98], [175] to advance the efficiency and effectiveness of model compression. In terms of multi-layer distillation, Javed *et al.* [174] propose an efficient KD method that utilizes multi-layer feature distillation, where a single layer in the student network is supervised by multiple layers from DNNs acting as teachers. In terms of multi-step distillation, Jiao *et al.* in [175] design a novel framework *TinyBERT* to distill knowledge from LLMs for efficient natural language understanding. *TinyBERT* adopted a two-stage learning framework to ensure it captures both general-domain and task-specific knowledge from BERT. To integrate quantification and distillation, Kim *et al.* in [177] delves into KD for enhancing QAT of large Transformer encoders like BERT. It highlights the limitations of previous KD methods in recovering self-attention information and introduces two novel KD approaches, attention-map and attention-output losses. These endeavors underscore the dynamic nature of KD methodologies, constantly pushing the boundaries of model optimization in various domains.

In summary, KD can migrate the performance of LLMs to smaller models on edge devices without sacrificing too much performance. This helps to fully leverage the power of LLMs in edge-cloud scenarios while avoiding excessive calculation and memory consumption.

V. LLMs Driven Edge-cloud Computing

In this section, we will discuss LLMs driven edge-cloud computing (i.e., *LLMs4Edges*). LLMs provide powerful support for diverse edge scenes through data generation, decision-making and personalized services. Specifically, data generation can expand information sources at network edges, decision-making optimizes real-time responsiveness, and personalized services enhance user experience.

A. LLMs for Data Generation at Edges

Deploying AIGC services on edge devices can significantly reduce transmission delay, enabling low-latency real-time response. This is important for applications that require fast interaction, such as augmented/virtual reality (AR/VR). LLMs can generate customized multi-modal content based on user needs and preferences at network edges [178]. Fig. 13 shows the LLMs empower edge networks from the aspect of text, image, video and scene generation.

Text Generation. LLMs play a key role in text generation applications in edge-cloud networks. They can generate natural language text locally on edge devices for use in intelligent chatbots, voice assistants, and localized automated text processing. Some typical application scenarios for text generation include: video-to-text, image-to-text or question-answering. In video-to-text, Alibaba Group proposes *Video-LLaMA* framework [179] that enables LLMs to understand

TABLE VI
SUMMARY OF PTQ AND QAT FOR LLMS

	Name/Ref	Scenarios	Main ideas	Performance improvements
Post-Training Quantization (PTQ)	Auto-Split [159]	Edge-cloud networks	Jointly apply model splitting along with PTQ to edge DNN	Auto-Split can achieve 7% faster and 20× smaller in edge DNN model size compared to QDMP + mixed precision algorithm
	QuantPipe [163]	MEC	Employ adaptive PTQ to compress tensors with dynamic bandwidth changes in edge environments	Improve accuracy under 2-bit quantization by 15.85% on ImageNet
	FPTQ [165]	\	Combine the benefits of 4-bit weight quantization and 8-bit matrix computation	Achieve state-of-the-art quantized performance on standard benchmarks like BLOOM, LLaMA, and LLaMA-2
Quantization -Aware Training (QAT)	Octo [160]	MEC	Enable on-device learning by 8-bit fixed-point (INT8) quantization in both forward and backward	Octo can achieve higher system performance over state-of-the-art quantization training methods on NVIDIA Jetson Xavier and HUAWEI Atlas 200DK
	LLM-QAT [167]	\	Propose a novel approach for QAT of LLMs by employing data-free distillation	Have large improvements in low-bit settings with LLaMA models of sizes 7B, 13B, and 30B.
	GPUSQ-TLM [168]	\	Prune and quantize TLMs by QAT to meet GPU acceleration constraints	Achieve state-of-the-art compression with minimal accuracy loss

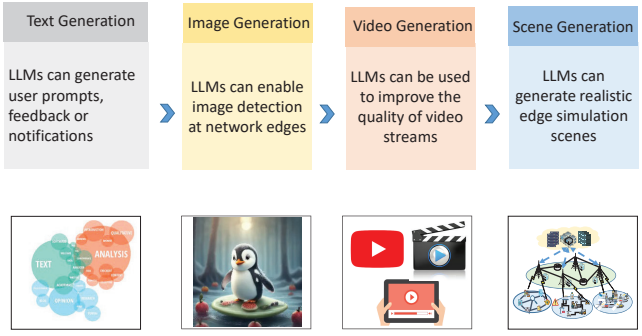


Fig. 13. LLMs for text, image, video and scene generation.

both visual and auditory content in videos. *Video-LLaMA* can perceive and comprehend video content, generating meaningful responses and corresponding text explanations, making it a promising prototype for audio-visual AI assistants at network edges. In question answering, Google Research exploits the effectiveness of using LLMs for image-based document question answering and provides valuable insights for future endeavours that emphasize the image-to-text and image content information in document question answering [180].

Apart from this, some studies also attempt to generate text from multi-modal input. This article [181] designs a novelty LLM framework *AnyMAL* that effectively processes a variety of multi-modal input modalities, such as text, image, video, audio, and sensor data, and generates textual responses. *AnyMAL* builds on the text-based inference capabilities of advanced LLMs like *LLaMA-2* and utilizes a pre-trained aligner module to unify modality-specific signals into a joint textual space. In conclusion, the use of LLMs in edge computing networks not only enhances text generation capabilities but also opens avenues for advanced multi-modal processing.

Image Generation. LLMs can be used for image generation (such as landscape synthesis, art generation, etc.) at network edges, which can achieve high-quality image creation and improve QoS/QoE. This helps drive the development of vision applications in edge-cloud networks. In text-to-image, this article [182] presents a method for integrating LLMs with image encoder and decoder models. The proposed approach effectively utilizes an efficient mapping network to translate LLM's text representations into the visual model's embedding space, leading to improved text-to-image generation flexibility performance. Lian *et al.* [183] address the challenge of complex prompts in text-to-image diffusion models. It introduces a two-stage approach, where LLM generates a scene layout in the first stage based on a given prompt. In the second stage, a controller guides an off-the-shelf diffusion model for image generation, significantly improving accuracy in generating images for diverse prompts. Baek *et al.*in [184] propose *PromptCrafter* to facilitate the step-by-step crafting of prompts for text-to-image generation models. Users can iteratively explore the model's capabilities, clarify their intent, and refine prompts through responses generated by LLMs.

In [185], *UniControl* is designed to handle a wide range of condition-to-image tasks while accommodating arbitrary language prompts. It achieves pixel-level precision in image generation, where visual conditions impact structures, and language prompts guide style and context. This article [186] proposes a novel approach that uses LLMs to extract critical components from text, including object bounding boxes, individual object descriptions, and background context to address challenges in text-to-image generation with complex and detailed prompts. In short, leveraging LLMs for image generation at edges holds the potential to enhance the quality of visual content creation while improving QoS/QoE for mobile users.

Video Generation. OpenAI releases the LLM framework *Sora* that can create realistic and imaginative scenes from text instructions. That means LLMs can be used for video gener-

ation [187], [188] and this will help deliver a smarter video experience for mobile users. The authors in [187] propose a video generation framework *VideoDirectorGPT* for generating consistent multi-scene videos. This framework uses LLM for video content planning, generating scene descriptions, entity layouts, backgrounds, and consistency groupings based on text prompts. Hong *et al.* in [189] presents *DirecT2V* for text-to-video generation by leveraging LLMs to generate frame-level descriptions from abstract user prompts and using LLM directors to separate prompts for each frame. Apart from this, Shen *et al.* in [190] design *Data Player* for automating the creation of dynamic data videos. *Data Player* achieves this by establishing connections between visualizations and textual input, extracting data from visuals, and leveraging LLMs to create semantic links between text and visuals. Exploring the intersection of edge computing and video generation, these innovative frameworks, such as *VideoDirectorGPT*, *DirecT2V*, and *Data Player*, showcase the potential of LLMs in transforming textual prompts into compelling visual narratives.

Multi-modal Data Generation. LLMs can process large-scale multi-modal data, including images, videos, text, and audio at network edges. They provide comprehensive analytics and insights for applications (such as intelligent monitoring and media processing). This helps achieve more intelligent and efficient data processing and applications, and promotes the development of edge-cloud networks. We list some multi-modal data generation studies as follows: Robert *et al.* in [191] design an any-to-any multi-modal framework *NExT-GPT* to understand and generate content in various modalities(i.e., text, images, videos, and audio). *NExT-GPT* achieves this by connecting an LLM with multi-modal adaptors and different diffusion decoders, making it a versatile system for handling arbitrary combinations of modalities. The authors in [192] achieve multi-modal image and text generation by using LLMs and propose a novel approach named *MiniGPT-5* to harmonize vision and language outputs.

Furthermore, in [65], the authors design a Composable Diffusion (*CoDi*) framework that is capable of producing various combinations of output modalities simultaneously. *CoDi* is not constrained to specific input types like text or images. It achieves this by aligning modalities in both input and output spaces, enabling it to condition on any input combination and generate any set of modalities, even when absent from training data. Lyu *et al.* in [66] presents a pioneering multi-modal LLM framework Macaw-LLM that seamlessly integrates visual, audio, and text data by comprising three key components: modality module, cognitive module and alignment module. These innovative works have demonstrated that the utilization of LLMs for multi-modal data generation can enhance the capacity to process diverse data types, which may foster advanced analytics and applications at the network edges.

Scene Generation. LLMs exhibit the capability to generate a myriad of scenes tailored for edge networks. Through real-time sensor data analysis, LLMs possess the ability to comprehend intricate environmental cues, subsequently producing immersive virtual scenes or simulation environments [193]. These generated scenarios find application in diverse fields, including AR/VR, autonomous driving [194], and the emerging metaverse [195]. This helps enhance perception and improve the intelligence and interactivity of edge networks. This article [196] proposes a novel approach *3D-LLMs*, aiming to incorporate the 3D physical world into LLMs. *3D-LLMs* are designed to handle multi-modal data and can perform a wide range of 3D-related tasks. This work opens up new possibilities for bridging LLMs with scene generation of edge-cloud networks. Wang *et al.* design *GenSim* [193] for generating simulation tasks and leveraging LLMs to create diverse simulation environments automatically. These studies lay the foundation for scene generation empowered by LLMs.

Apart from this, this article presents *CTG++* [194], a scene-level conditional diffusion model for realistic and controllable traffic simulation in the context of an autonomous driving environment at network edges. Tang *et al.* in [195] design an interactive 3D generation system *LI3D* that leverages LLMs for interpreting 3D layouts. The *LI3D* system is also validated through multi-round interactions, demonstrating its effectiveness in 3D/2D generation and editing, with potential applications in the metaverse. In the realm of edge scene generation, these advancements, including *3D-LLMs*,*GenSim* and *CTG++*, mark a transformative stride towards creating immersive and intelligent environments. The integration of LLMs enhances the adaptability and realism of simulations, paving the way for innovative applications and richer user experiences in the evolving landscape of edge-cloud networks. We summarize the model, scenarios, parameter scale, contributions or Performance and environment of text, image, video, multi-modal data and scene generation in Table VII.

B. LLMs for Decision-Making at Edges

LLMs have been widely used as general decision-making models at edge networks due to their advanced inference capabilities and rich world knowledge [197]. There are often numerous scenarios (e.g., computation offloading [22], content caching, model coordination [27], energy consumption reduction [198], etc.) at network edges that require LLMs to make decisions. These scenarios often need to quickly analyze complex data and make decisions based on analysis results.

Specifically, for computation offloading, Dong *et al.* propose LLM-Based offloading framework *LAMBO* [22], which combines LLMs with MEC for edge-enhanced intelligence task offloading systems. *LAMBO* includes different components of input embedding, asymmetric encoder-decoder model, actor-critic-based DRL, and active learning from expert feedback. In terms of model coordination, the authors in [27] investigate the orchestration of edge AI models, GPT is used to handle user requests, decompose them into sub-tasks, and then allocate these sub-tasks to the relevant edge devices and AI models for execution. To reduce resource consumption, Zou *et al.* [198] consider a scenario with multiple intelligent terminals to reduce the total power consumption in mobile edge computing networks. The authors evaluate the capability of on-device LLMs in solving a wireless communication problem via GPT-4 and show their potential in enabling end-to-end autonomous edge networks. These frameworks [22], [27] or applications [198] necessitate swift decision-making based on intricate data

TABLE VII
SUMMARY OF TEXT, IMAGE, VIDEO, MULTI-MODAL DATA AND SCENE GENERATION OF LLMS FOR EDGE-CLOUD NETWORKS

	Name/Ref	Model	Scenarios	Parameter scale	Contributions or Performance	Environment
Text Generation	Video-LLaMA [179]	LLaMA	Video-to-text	7B/13B	Generate meaningful responses and corresponding text explanation based on video content.	8×A100-80G GPU
	AnyMAL [181]	LLaMA	Multi-modal -to-text	70B	Effectively process multi-modal input modalities and generates textual responses.	80GB VRAM GPU
Image Generation	UniControl [185]	\	Condition -to-image	1.44B	Achieve pixel-level precision in image generation, and language prompts guide style and context.	Nvidia A100-40G GPU
	Scene Blueprints [186]	GPT	Text-to-image	\	Address challenges in text-to-image generation with complex and detailed prompts.	Nvidia A100-40GB GPU
Video Generation	Video-Director -GPT [187]	GPT	Text-to-video	\	Use LLMs for video content planning, generating scene descriptions, entity layouts, backgrounds, and consistency groupings based on a text prompt	8 × A6000 GPUs-48GB
	DirecT2V [189]	GPT	Text-to-video	\	Leverage LLMs to generate frame-level descriptions from user prompts and separate prompts for each frame	Nvidia RTX 3090 GPU
Multi-modal Data Generation	NExT-GPT [191]	GPT	Multi-modal	11.3B	Propose an any-to-any multi-modal framework to understand and generate content in various modalities, including text, images, videos, and audio.	\
	MiniGPT-5 [192]	LLaMA	Multi-modal	$\geq$ 7B	Address the challenge of multi-modal image and text generation	4× A6000 GPUs
Scene Generation	LI3D [195]	LLaMA/GPT	3D/2D scene	$\geq$ 13B	Design a language-guided interactive 3D/2D generation and edition system	NVIDIA RTX3090
	3D-LLM [196]	GPT	3D scene	$\geq$ 11.7B	Incorporate the 3D physical world into LLMs and opens up new possibilities for bridging language models with scene generation	64 ×NVIDIA Tesla V100S

analysis, showcasing the adaptability and versatility of LLMs in edge computing environments.

Another typical decision-making scenario is the application of LLMs in the decision-making of agents [57], [197], [199]–[201]. Xi *et al.* in [57] makes a significant contribution by presenting LLM-based agents that encapsulate components of the brain, perception, and action. This comprehensive approach extends the application domain to encompass single-agent, multi-agent, and human-agent decision-making scenarios, providing valuable insights. To enhance problem-solving in IoT applications, this paper [197] proposes the *GPT-in-the-loop* approach, which combines LLMs like GPT-4 with multi-agent systems. The proposed *GPT-in-the-loop* approach can enable agents to achieve superior decision-making and adaptability without extensive training.

Furthermore, Chen *et al.* [199] exploits the potential of multi-modal LLMs (*MLLMs*) in improving embodied decision-making for agents. The authors prove that *MLLMs* has stronger visual understanding and inference capabilities. The authors in [200] design *OlaGPT* provides templates for "Chain-of-Thought" and design a decision-making mechanism, aiming at enhancing LLMs problem-solving capabilities by simulating aspects of human cognition. This study designs a novel approach *LLM-Planner* [201] that leverages LLMs as a planner for embodied agents capable of executing complex tasks in visually perceived environments. *LLM-Planner* uses less than 0.5% of paired training data and addresses the limitations of data cost and sample efficiency. These studies collectively demonstrate the versatility and efficacy of incorporating LLMs into various decision-making scenarios.

In terms of the framework design of agent decision-making, the authors in [202] propose a novel multi-agent decision transformer framework (*MADT*). *MADT* can effectively integrate sequence modelling with offline and online tasks, offering generalizable decisions that transfer across different types of agents and scenarios. Apart from this, DeepMind design *Gato* [203] to unify single-agent decision-making tasks, multi-round dialogue and picture-text generation tasks into a transformer-based architecture, and achieve excellent performance on 604 different tasks, and can solve some simple DRL decision-making problem. To implement intelligent decision-making (IDM) systems, Wen *et al.* in [204] designs the framework DigitalBrain (*DB1*) in dynamic real-world environments. Specifically, the authors propose a foundation decision model using the transformer to tackle decision-making tasks, enabling broader IDM applications. From the framework design perspective, these advancements collectively contribute to the evolving landscape of intelligent decision-making systems.

Case Study: To verify the powerful decision-making ability of LLMs, we design a case study of content caching

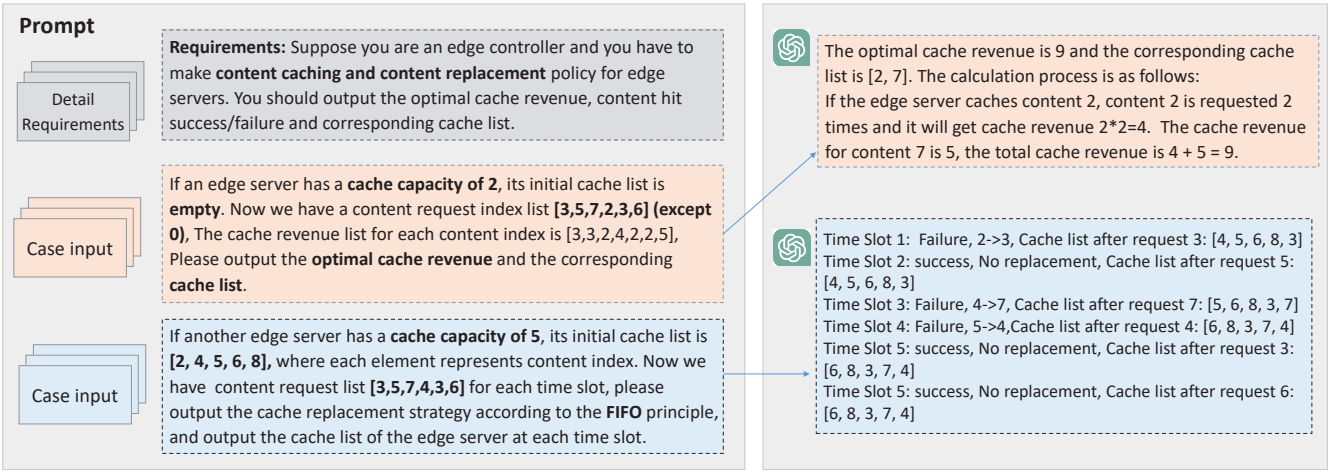


Fig. 14. A case study of LLM for content caching and replacement of edge networks. The grey text denotes the task description, and the orange text and blue text represent the scenario of content caching and replacement, respectively.

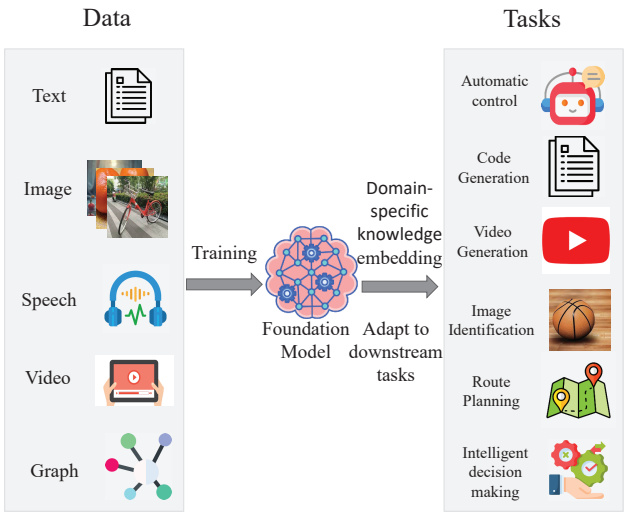


Fig. 15. Adapt LLMs to downstream personalized services with domain-specific knowledge embedding.

and replacement of edge networks based on GPT-4/3.5. As shown in Fig. 14, the grey text denotes the task description, and the orange text and blue text represent the scenario of content caching and replacement, respectively. We enable GPT to function as an edge controller, and make content caching/replacement policies for edge servers. In experiments with content caching, we set the cache capacity of the edge server as a constant value and give the content request index at each time slot. With this given information, the edge controller outputs the optimal cache revenue and the corresponding cache list. We find that if we give GPT accurate prompt words, it can make precise content caching strategies for me. Similarly, we provide the GPT with cache capacity and content request list, the GPT also returns the precise content replacement strategies and cache list at each time slot.

C. LLMs for Personalized Service at Edges

Deploying LLMs at network edges allows for the harnessing of their capabilities in delivering personalized services and generating personalization models [205], [206].

One feasible way for personalized service is to embed domain-specific knowledge into LLMs [207]–[209], as shown in Fig. 15. This approach allows LLMs to seamlessly integrate domain expertise, enhancing their ability to provide personalized services. By embedding specific knowledge relevant to the domain's context, these LLMs can cater to individualized needs, delivering more precise and tailored assistance in real-time. This article [209] proposes *KITLM*, a novel approach for integrating domain-specific knowledge into LLMs to enhance their performance in question-answering tasks. By infusing relevant knowledge, *KITLM* achieves a significant performance boost while reducing the model's size requirements. This approach provides the possibility to implement personalized services at network edges.

In industrial applications scenarios, Amazon [210] proposes an approach named *domain prompts* for efficient domain adaptation of automatic speech recognition systems. The proposed approach trains a small set of domain embedding parameters to prime transformer-based LLMs for a specific domain, which update only 0.02% of the base LLM parameters without altering the base model architecture. Wen *et al.* propose *MindMap* [211] to enhance LLMs incorporation of new knowledge and intelligent decision-making. The proposed prompting pipeline integrates knowledge graphs to engage LLMs with up-to-date knowledge, facilitating reasoning pathways generation. In the context of injecting domain-specific knowledge into LLMs, these approaches [209]–[211] contribute to a paradigm shift in advancing model capabilities.

Besides, the collaborative approach [212] between LLMs and small models can also be used to provide personalized services. This collaboration paradigm leverages the strengths of both LLMs and smaller models to offer personalized services effectively on mobile devices. The amalgamation of comprehensive knowledge from LLMs with the computa-

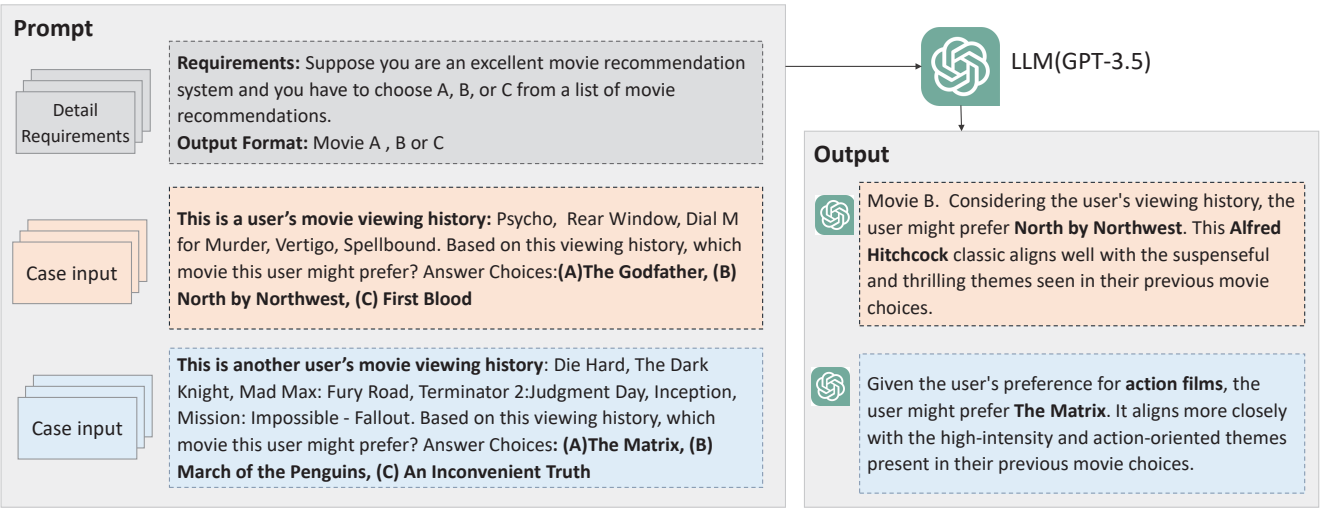


Fig. 16. A case study of LLM for personalized service at network edges. The grey text denotes the task description and requirements, and the orange text and blue text represent two movie recommendation scenes, respectively.

tional efficiency of smaller models ensures a responsive and personalized user experience, optimizing resource utilization for mobile users. Vernikos *et al.* in [212] propose approach involves using a smaller model, LM-corrector (*LMCor*), to refine LLM-generated outputs by ranking, combining, and rewriting candidates. This collaboration between large and small models shows that even a small model *LMCor* significantly can also enhance the performance of LLMs across various tasks, and offer robustness to different prompts for improved performance. Another work, this article [213] integrates generative LLMs with domain-specific small models for personalized recommendation. The proposed approach introduces an information-sharing module as a bridge, facilitating collaborative training between LLMs and domain-specific small models. This collaborative approach provides convenience for LLMs to adapt to downstream tasks.

In terms of collaboration between LLMs and small models at edge-cloud networks, the authors in [27] propose an autonomous edge AI system with a hierarchical architecture that deploys GPT in cloud servers and other AI models on edge devices or edge servers. The system efficiently coordinates AI models to meet users' personalized demands and generates code for training new models through edge FL.

Recommendation systems are another way to implement personalized services at the edge-cloud networks. It leverages user preferences and behavior patterns to generate personalized recommendations, tailoring content or services to individual needs. By deploying LLMs at network edges, personalized content recommendations and context-aware interactions can be seamlessly integrated [22], [214]. Recently, the integration between edge-cloud computing and recommendation systems has been fully studied [215], [216]. Edge-cloud computing with LLMs provides new possibilities for personalized services for mobile users. LLMs can implement more intelligent and personalized recommendations on edge devices and provide users with customized experiences [217].

We list some studies about the integration between LLMs and recommendation systems as follows. This paper [218] exploits the integration of ChatGPT/GPT4 in Recommendation systems to overcome limitations faced by traditional DNN models. It provides a comprehensive review of LLM-empowered Recommendation systems, covering aspects like pre-training, fine-tuning, and prompting. In [219], the authors propose *LLM-Rec* and exploit various prompting strategies to enhance personalized recommendation performance by using LLMs. The authors in [220] propose *InteRecAgent* that combines the strengths of LLMs and Recommendation models to create an interactive Recommendation system. By using LLMs as the brain and Recommendation models as tools, it bridges the gap between domain-specific recommendations and versatile conversational interactions. In addition, this article [221] proposes a novel approach for personalized recommendations using LLMs by extracting and fusing knowledge from heterogeneous user behavior information.

In addition, some scholars have also conducted research on LLMs promoting edge personalized services from other aspects [84], [222]. Yang *et al.* design a novel edge-cloud cooperative system *EdgeFM* [222] for effectively utilizing foundation models (FMs) on resource-limited IoT devices. *EdgeFM* selectively uploads unlabeled data to query FMs on the cloud, customizing knowledge for edge models. Evaluation results demonstrate that *EdgeFM* reduces latency by up to 3.2× and achieves a 34.3% accuracy improvement compared with baseline schemes. Chen *et al.* propose a native AI network architecture *NetGPT* [84] for provisioning personalized generative services. *NetGPT* efficiently deploy LLMs (e.g., GPT-2 and LLaMA) based on computing capacity in collaborative edge-cloud networks. In addition, *NetGPT* emphasizes the integration of communication and computing resources and highlights its potential for intelligent network management and orchestration by harnessing edge LLMs' trend prediction and intent inference capabilities. These innovative approaches in LLMs research contribute significantly to the evolution of personalized services at edge computing networks.

Case Study: We demonstrate the powerful ability of LLMs to provide personalized services through movie recommendations. As shown in Fig. 16, the grey text denotes the task description and requirements, and the orange text and blue text represent two movie recommendation scenes, respectively. GPT provides insights into user likes and preferences to tailor unique movie recommendations to each user, improving the overall user experience. By analyzing the user's movie viewing history, rating records, preference tags and other information, they can accurately predict the types of movies that the user may like, making the recommendation list more in line with the tastes of individual users, and providing users with more personalized and satisfying movie viewing. In the first scenario, based on the user's movie viewing list, GPT found that the user is likely to like Hitchcock's movies, so it recommend North by Northwest. In the second scenario, GPT recommended the action movie "The Matrix" based on the type of movies the user liked to watch recently.

VI. Applications of LLMs at Edges

In this section, we will introduce the applications of LLMs at the edges of networks, as shown in Fig. 17. We discuss autonomous driving, smart healthcare, smart factory and intelligent anomaly detection systems with LLMs.

A. Autonomous Driving

LLMs can optimize the perception, decision-making [223], [224], optimize vehicle control [225], scene understanding [226], route planning [227], [228] and improve driving experience [229] for autonomous driving at network edges [230]. Specifically, Sha *et al.* in [223] use LLMs as decision-makers in autonomous driving systems to enhance understanding of high-level information and rare events, while ensuring interpretability. The authors propose cognitive pathways for comprehensive inference and translate LLM decisions into actionable driving commands. In [225], the authors develop an interpretable end-to-end autonomous driving system *DriveGPT4* that utilizes LLMs to interpret vehicle actions and answer user questions for enhanced interaction. It also predicts low-level control signals, all facilitated by a custom visual instruction tuning dataset. The authors in [227] design *GPT-DRIVER* that transform the OpenAI GPT-3.5 model into a motion planner for autonomous vehicles. Through a prompt fine-tuning strategy, *GPT-DRIVER* can provide highly precise trajectory coordinates and explain its decision-making process. In [228], the authors propose a unified autonomous driving (*UniAD*) framework, which combines perception, and route planning into one comprehensive LLM.

B. Smart Healthcare

The utilization of LLMs plays a crucial role in smart healthcare by facilitating real-time analysis of medical data [231], [232], safety assessment [233], and diagnostic assistance [234], [235]. These applications collectively contribute to enhancing the efficiency and quality of medical care in edge networks [236]. Li *et al.* in [231] adopt LLMs to achieve automatic segmentation and recognition of various types of images in medical image analysis, thus achieving the goal of assisting doctors in making sensitive and accurate diagnoses. The authors in [232] propose a multi-modal image translation model based on two steps. Specifically, the first step is to detect abnormal areas, and in the second step, LLM generates corresponding text and translates images into an interpretable language description. Apart from this, the study [234] designs a plug-and-play medical dialogue system *PlugMed* that enhances LLMs for medical conversations without fine-tuning. The study [235] addresses the limitations of LLMs in medical question answering. It proposes a model editing approach that utilizes in-context learning to enhance LLM responses without requiring fine-tuning. The proposed strategy involves retrieving medical facts from external knowledge bases and incorporating them into the query prompt for LLMs.

C. Smart Factory

LLMs can be widely used in smart factories to optimize production planning, real-time monitoring and achieve predictive maintenance at network edges. Freire *et al.* in [237] exploit the potential of harnessing LLMs, such as GPT-3.5, for cognitive assistants in smart factory to enhance knowledge transfer and worker performance in manufacturing [71]. This is a meaningful attempt to apply LLMs to intelligent manufacturing. In addition, LLMs are also used in the control of factory robots (e.g., planning control [238], [239], human-robot interaction [240]). This article [238] design *LLM-Brain* that utilizes LLMs as a unified robotic brain, integrating egocentric memory and control in embodied AI systems. The framework employs multi-modal LLMs for robotic tasks, facilitating communication across perception, planning, control, and memory through natural language dialogues. The authors in [239] enhance autonomous robotic manipulation by integrating LLMs for logical inference. The proposed framework leverages LLMs to convert high-level commands into executable motion functions, combining them with YOLO-based environmental perception. This allows robots to make informed decisions based on provided commands autonomously.

D. Intelligent Anomaly Detection System

In edge environments, diverse anomalies like semantic aberrations [241], industrial anomalies [242], [243] and network intrusions [244] pose huge challenges. LLMs prove effective in addressing these issues through advanced pattern recognition and anomaly detection capabilities. This article [241] exploits the application of LLMs for semantic anomaly detection in robotic systems, addressing edge cases or failures caused by deficiencies in semantic reasoning. The authors design *HuntGPT* [244] that incorporates a GPT-3.5 Turbo and random forest classifier, delivering robust and explainable intrusion detection, and offering an interactive AI solution for incident responders. Apart from this, there is also some research on anomaly detection based on system logs. The study introduces a log-based anomaly detection framework *LogGPT* [245] that utilizes the language interpretation capabilities of ChatGPT to address challenges in high-dimensional, noisy log data. This

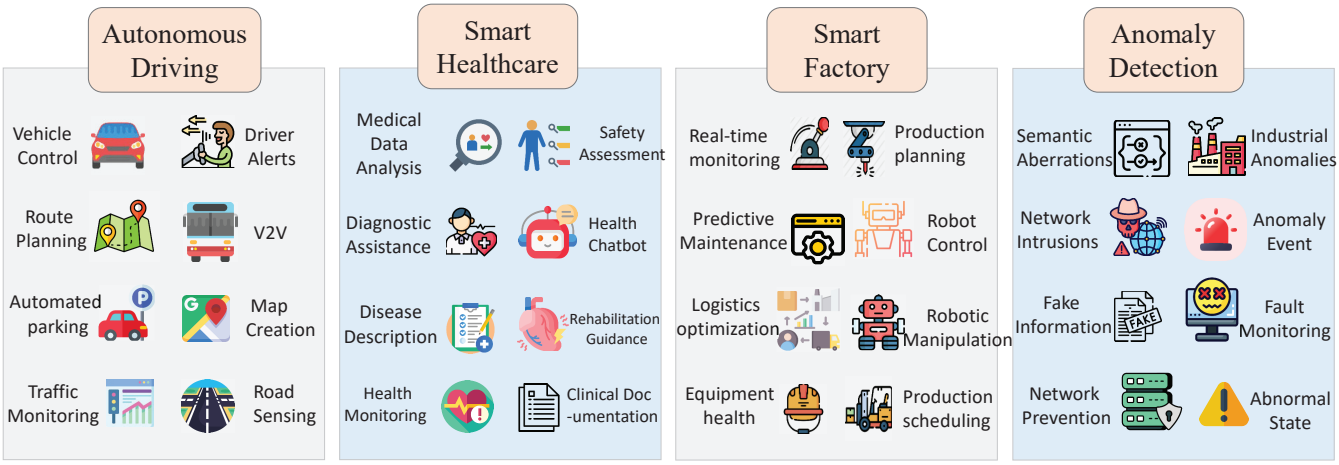


Fig. 17. Applications of autonomous driving, smart healthcare, smart factory and intelligent anomaly detection system with LLMs at edge-cloud networks.

article [246] exploits the potential of utilizing LLMs for log-based anomaly detection in parallel file systems, addressing issues in traditional techniques such as the reliance on expert-labeled logs. Therefore, LLMs have emerged as essential tools in defect detection, showcasing notable effectiveness within this domain.

VII. Research Challenges And Future Directions

In this section, we will discuss the research challenges and future directions from the aspect of model collaboration, adapting LLMs to edge-diverse scenarios and computing power resource optimization of LLMs at edge-cloud networks.

A. Model Collaboration

In the context of collaboration between LLMs on cloud servers and small models at network edges, model collaboration faces a series of challenges. These challenges involve many aspects such as heterogeneous model design, communication overhead, real-time and low-latency requirements.

First, regarding heterogeneous model design, there may be architectural and feature differences between LLMs and small models on edge devices [213]. Many researchers use methods such as transfer learning (TL) and feature distillation to transfer knowledge between LLMs in cloud servers and small models at network edges. However, heterogeneity between different models can lead to differences that are difficult to resolve, requiring more sophisticated TL methods or feature distillation techniques. In some scenarios, a small model may not capture all the information in a larger model.

Second, from the perspective of communication overhead, transmitting data or model parameters between LLMs and small models can result in an unbearable communication overhead, especially in resource-constrained edge environments with limited network bandwidth. Some authors use model compression techniques, including model pruning [247], quantization [161], distillation [173], etc., to reduce model size and computational complexity. However, these model compression strategies may lead to model performance degradation, and thus a trade-off needs to be carefully investigated between model compression and training/inference performance. Additionally, some common model compression methods may not be suitable for all types of models.

Finally, considering real-time and low-latency requirements, many edge applications have strict delay requirements and the computational complexity of LLMs can lead to extended inference latency. Some technology companies use hardware accelerators, optimization algorithms, or selective loading and unloading of models to meet real-time and low-latency needs. Despite employing hardware acceleration and algorithm optimization, the computational complexity of LLMs may pose challenges in meeting strict real-time and low-latency requirements under certain extreme conditions.

Therefore, the future research direction of LLMs and edge-cloud computing can be considered from the following three aspects: i) Collaborative optimization and balance: How LLMs and small models be effectively coordinated between cloud servers and edge devices to ensure an effective balance between performance and resource utilization; ii) Performance optimization: Transferring data between the cloud servers and edge devices may be affected by communication bandwidth, especially when edge network conditions are poor. How to achieve seamless integration of LLMs and small models to provide an end-to-end smooth user experience, while considering the heterogeneous model design; iii) Model design under resource constraints: Edge devices are often constrained by computing and storage resources. How to design small models that accommodate these constraints and ensure their effectiveness in working together with LLMs in the cloud.

B. Adapt LLMs to Edge Diverse Scenarios

When applying LLMs to edge diverse scenarios, there exist some potential research challenges: data heterogeneity and scarcity, as well as the long-term adaptability of LLMs.

To solve the problem of data heterogeneity and scarcity, some researchers have designed domain adaptation methods in the training process to make LLMs better adapt to the specific data distribution at different edge scenarios. However, edge scenarios may involve diverse fields, making domain adaptation more difficult, especially when the data distribution

changes rapidly. In addition, some studies have also tried to use meta-learning methods [248] to enable LLMs to adapt to new tasks or data distributions, thereby mitigating the impact of data scarcity. However, meta-learning methods may also overfit a specific task in some cases, resulting in insufficient generalization in real-world scenarios.

To address the challenge of ensuring the long-term adaptability of LLMs in diverse edge scenarios, some researchers have exploited various approaches. One key strategy involves the implementation of online learning techniques, allowing the model to continuously receive and adapt to new data over Qtime. This enables the LLMs to stay relevant and effective in dynamic edge environments where the data landscape evolves. However, in online learning, the model may forget previously learned knowledge. Therefore, it is necessary to design effective memory mechanisms or incremental learning algorithms. Additionally, data distribution may change over time, leading to concept drift. It is necessary to design methods to detect concept drift to ensure long-term adaptability.

Therefore, investigating the issue of data heterogeneity and scarcity, and the long-term adaptability of LLMs at edge-cloud networks is becoming a future research direction. Specifically, i) Data heterogeneity and scarcity: Data in different edge scenarios may be heterogeneous [249], including structured and unstructured data, real-time and offline data, etc. How can LLMs better adapt to this heterogeneity and improve their generalization ability; Besides, in some edge scenarios, there is very limited training data [250]. How LLMs can be tailored to mitigate the impact of data scarcity in these specific edge scenarios, ensuring robust performance. ii) Ensuring long-term adaptability: In a dynamic landscape of emerging edge scenarios, it is imperative to design LLMs that exhibit enduring adaptability. The rapid evolution of edge-cloud computing imposes new challenges and opportunities, requiring LLMs to stay relevant over the long term. How to design LLMs to maintain long-term adaptability and cope with the rapid evolution of emerging edge scenarios.

C. Computing Power Resource Optimization

LLMs face many challenges in edge-cloud networks, involving computing power supply, adjustment and trading.

First, unbalanced computing power supply: the computing capabilities of edge devices may vary greatly. That is, some devices may have abundant computing resources while others may be restricted. Some studies use dynamic allocation mechanisms to intelligently adjust computing power allocation by monitoring the status and load of edge devices [15]. However, in large-scale edge networks, real-time monitoring and adjustment of computing power allocation may lead to large system overhead. At the same time, there may be some devices that are unwilling or unable to share their real-time status, resulting in incomplete information.

Secondly, dynamic computing power adjustment: the edge environment may change at any time, and how to dynamically adjust computing power to adapt to different network conditions and device status is challenging. Some scholars have developed adaptive algorithms that can automatically adjust the allocated computing power according to changes in the edge environment [125], [128], such as dynamic adjustments based on task generation rate, device load, etc. However, rapid changes in edge environments may result in algorithm insensitivity or untimely adjustment processes. Besides, too frequent adjustments may cause resource jitter and instability.

Thirdly, unfair computing power trading: There may be multiple entities providing computing power in edge-cloud networks, and the trading and management of computing power involve the collaboration of multiple parties. BC technology may bring large delays and high computational overhead [251]. The design of smart contracts also requires caution, as it is difficult to ensure completely fair, transparent and efficient trading. In addition, in terms of the flexibility and elasticity of computing power trading, LLMs may need to run on different devices and network conditions. How to achieve flexibility and elasticity in computing power trading is also a challenge. Containerization and virtualization technologies can enable some LLMs to run on different edge devices [54]. However, it also brings certain performance overhead, and not all edge devices can easily support these technologies.

Based on these, we need to design computing power optimization strategies systematically. We can conduct research from aspects of the architectural level and sustainability level in the future. i) New architecture and algorithm design: Faced with the increasing demand for computing power, new computing power architecture and trading algorithm design have become crucial. It is significant to explore some possible innovations in computing power network architecture in the future to improve the computing efficiency of LLMs. ii) Energy efficiency and sustainability: The increasing size of LLMs also creates challenges in terms of computing power efficiency and sustainability. The number of parameters of some LLMs has reached tens of billions or even hundreds of billions (e.g., LLAMA2-70B and GLM-130B). Reducing the consumption of computing resources while maintaining high performance is an urgent problem that needs to be solved.

VIII. Conclsion

In this survey, we provide a comprehensive overview of the integration of edge-cloud computing and LLMs, seeking to harness their synergistic potential for a harmonized and advanced computing methodology. Specifically, We introduce the characteristics, development and life-cycle of LLMs at first. Then, we discuss the paradigms of edge-cloud computing and edge AI hardware for the training and inference of LLMs. After that, we focus on exploring the bidirectional symbiotic relationship between edge-cloud computing and LLMs from two key aspects: edge-cloud computing empowered LLMs, i.e., *Edge4LLMs* and LLMs driven edge-cloud computing, i.e., *LLMs4Edge*. Then, we give some practical applications of LLMs in different edge-cloud scenarios. Finally, we highlight open issues and future directions to implement intelligent integration of edge-cloud computing and LLMs. We hope that this article will motivate further research on the synergy of edge-cloud computing and LLMs, and offer some guidance in future studies.

REFERENCES

[1] N. Fei, Z. Lu, Y. Gao, G. Yang, Y. Huo, J. Wen, H. Lu, R. Song, X. Gao, T. Xiang *et al.*, "Towards artificial general intelligence via a multimodal foundation model," *Nature Communications*, vol. 13, no. 1, p. 3094, Jun. 2022.

[2] Y. Yue and J. Z. Shyu, "An overview of research on human-centered design in the development of artificial general intelligence," *arXiv preprint arXiv:2309.12352*, Aug. 2023.

[3] F. Dou, J. Ye, G. Yuan, Q. Lu, W. Niu, H. Sun, L. Guan, G. Lu, G. Mai, N. Liu *et al.*, "Towards artificial general intelligence (agi) in the internet of things (iot): Opportunities and challenges," *arXiv preprint arXiv:2309.07438*, Sep. 2023.

[4] S. Altman, "Planning for agi and beyond," Feb. 2023. [Online]. Available: https://openai.com/blog/planning-for-agi-and-beyond

[5] M. Moor, O. Banerjee, Z. S. H. Abad, H. M. Krumholz, J. Leskovec, E. J. Topol, and P. Rajpurkar, "Foundation models for generalist medical artificial intelligence," *Nature*, vol. 616, no. 7956, pp. 259–265, Apr. 2023.

[6] E. Latif, G. Mai, M. Nyaaba, X. Wu, N. Liu, G. Lu, S. Li, T. Liu, and X. Zhai, "Artificial general intelligence (agi) for education," *arXiv preprint arXiv:2304.12479*, Nov. 2023.

[7] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg *et al.*, "Sparks of artificial general intelligence: Early experiments with gpt-4," *arXiv preprint arXiv:2303.12712*, Apr. 2023.

[8] Y. Chang, X. Wang, J. Wang, Y. Wu, K. Zhu, H. Chen, L. Yang, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *arXiv preprint arXiv:2307.03109*, Oct. 2023.

[9] C. Zhang, C. Zhang, C. Li, Y. Qiao, S. Zheng, S. K. Dam, M. Zhang, J. U. Kim, S. T. Kim, J. Choi *et al.*, "One small step for generative ai, one giant leap for agi: A complete survey on chatgpt in aigc era," *arXiv preprint arXiv:2304.06488*, Apr. 2023.

[10] OpenAI, "Gpt-4 technical report," Mar. 2023. [Online]. Available: https://cdn.openai.com/papers/gpt-4.pdf

[11] B. Munday, "Deploy and run llms at the edge," Oct. 2023. [Online]. Available: https://medium.com/getmodzy/deploy-and-run-llms-at-the-edge-90b8523f6d85

[12] J. Du, J. Jiang, J. Zheng, H. Zhang, D. Huang, and Y. Lu, "Improving computation and memory efficiency for real-world transformer inference on gpus," *ACM Trans. Arch. Code Opt.*, vol. 20, no. 4, pp. 1–22, Oct. 2023.

[13] M. S. Elbamby, C. Perfecto, C.-F. Liu, J. Park, S. Samarakoon, X. Chen, and M. Bennis, "Wireless edge computing with latency and reliability guarantees," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1717–1737, Aug. 2019.

[14] H. Li, X. Li, C. Sun, F. Fang, Q. Fan, X. Wang, and V. C. Leung, "Intelligent content caching and user association in mobile edge computing networks for smart cities," *IEEE Trans. Netw. Sci. Eng.*, 2023, doi:https://doi.org/10.1145/3597023.

[15] Z. Xu, D. Yang, C. Yin, J. Tang, Y. Wang, and G. Xue, "A co-scheduling framework for dnn models on mobile and edge devices with heterogeneous hardware," *IEEE Trans. Mobi. Comput.*, vol. 22, no. 3, pp. 1275–1288, Mar. 2023.

[16] N. Lin, H. Lu, X. Hu, J. Gao, M. Zhang, and X. Li, "When deep learning meets the edge: Auto-masking deep neural networks for efficient machine learning on edge devices," in *Proc. IEEE ICCD*, Nov. 2019, pp. 506–514.

[17] M. Li, Y. Li, Y. Tian, L. Jiang, and Q. Xu, "Appealnet: An efficient and highly-accurate edge/cloud collaborative architecture for dnn inference," in *ACM/IEEE Design Automation Conference (DAC)*, Dec. 2021, pp. 409–414.

[18] I. Kandaswamy, S. Farkya, Z. Daniels, G. van der Wal, A. Raghavan, Y. Zhang, J. Hu, M. Lomnitz, M. Isnardi, D. Zhang *et al.*, "Real-time hyper-dimensional reconfiguration at the edge using hardware accelerators," in *Proc. IEEE/CVF CVPR*, Jul. 2022, pp. 3610–3618.

[19] Z. Hao, G. Xu, Y. Luo, H. Hu, J. An, and S. Mao, "Multi-agent collaborative inference via dnn decoupling: Intermediate feature compression and edge learning," *IEEE Trans. Mobi. Comput.*, vol. 22, no. 10, pp. 6041–6055, Oct. 2023.

[20] S. Yang, Z. Zhang, C. Zhao, X. Song, S. Guo, and H. Li, "Cnnpc: End-edge-cloud collaborative cnn inference with joint model partition and compression," *IEEE Trans. Para. Distri. Sys.*, vol. 33, no. 12, pp. 4039–4056, Dec. 2022.

[21] S. Carreira, T. Marques, J. Ribeiro, and C. Grilo, "Revolutionizing mobile interaction: Enabling a 3 billion parameter gpt llm on mobile," *arXiv preprint arXiv:2310.01434*, Sep. 2023.

[22] L. Dong, F. Jiang, Y. Peng, K. Wang, K. Yang, C. Pan, and R. Schober, "Lambo: Large language model empowered edge intelligence," *arXiv preprint arXiv:2308.15078*, Aug. 2023.

[23] H. Woisetschläger, A. Isenko, S. Wang, R. Mayer, and H.-A. Jacobsen, "Federated fine-tuning of llms on the very edge: The good, the bad, the ugly," *arXiv preprint arXiv:2310.03150*, Oct. 2023.

[24] M. Xu, Y. Wu, D. Cai, X. Li, and S. Wang, "Federated fine-tuning of billion-sized language models across mobile devices," *arXiv preprint arXiv:2308.13894*, Aug. 2023.

[25] H. Wen, Y. Li, G. Liu, S. Zhao, T. Yu, T. J.-J. Li, S. Jiang, Y. Liu, Y. Zhang, and Y. Liu, "Empowering llm to use smartphone for intelligent task automation," *arXiv preprint arXiv:2308.15272*, Aug. 2023.

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

[27] Y. Shen, J. Shao, X. Zhang, Z. Lin, H. Pan, D. Li, J. Zhang, and K. B. Letaief, "Large language models empowered autonomous edge ai for connected intelligence," *arXiv preprint arXiv:2307.02779*, Jul. 2023.

[28] J. Yin, J. Dong, Y. Wang, C. D. Sa, and V. Kuleshov, "Modulora: Finetuning 3-bit llms on consumer gpus by integrating with modular quantizers," *arXiv preprint arXiv:2309.16119*, Sep. 2023.

[29] Z. Guo, R. Jin, C. Liu, Y. Huang, D. Shi, L. Yu, Y. Liu, J. Li, B. Xiong, D. Xiong *et al.*, "Evaluating large language models: A comprehensive survey," *arXiv preprint arXiv:2310.19736*, Nov. 2023.

[30] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin, and X. Hu, "Harnessing the power of llms in practice: A survey on chatgpt and beyond," *arXiv preprint arXiv:2304.13712*, Apr. 2023.

[31] E. L. Rimban, "Challenges and limitations of chatgpt and other large language models," *International Journal of Arts and Humanities*, vol. 4, no. 1, pp. 147–152, Apr. 2023.

[32] A. Borzunov, M. Ryabinin, A. Chumachenko, D. Baranchuk, T. Dettmers, Y. Belkada, P. Samygin, and C. Raffel, "Distributed inference and fine-tuning of large language models over the internet," *arXiv preprint arXiv:2312.08361*, Dec. 2023.

[33] A. M. Ghosh and K. Grolinger, "Edge-cloud computing for internet of things data analytics: Embedding intelligence in the edge with deep learning," *IEEE Trans. Indust. Inform.*, vol. 17, no. 3, pp. 2191–2200, Jul. 2020.

[34] T. Wang, L. Qiu, A. K. Sangaiah, A. Liu, M. Z. A. Bhuiyan, and Y. Ma, "Edge-computing-based trustworthy data collection model in the internet of things," *IEEE Internet of Things J.*, vol. 7, no. 5, pp. 4218–4227, Jan. 2020.

[35] X. Kong, Y. Wu, H. Wang, and F. Xia, "Edge computing for internet of everything: A survey," *IEEE Internet of Things J.*, vol. 9, no. 23, pp. 23 472–23 485, Aug. 2022.

[36] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys & Tutorials*, Sep. 2023, 10.1109/TSUSC.2023.3291365.

[37] G. Bao and P. Guo, "Federated learning in cloud-edge collaborative architecture: key technologies, applications and challenges," *J. Cloud Comput.*, vol. 11, no. 1, p. 94, Dec. 2022.

[38] C. Zhang, J. Xia, B. Yang, H. Puyang, W. Wang, R. Chen, I. E. Akkus, P. Aditya, and F. Yan, "Citadel: Protecting data privacy and model confidentiality for collaborative learning," in *Proc. ACM Symposium on Cloud Computing*, Nov. 2021, pp. 546–561.

[39] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge ai: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wire. Commun.*, vol. 19, no. 1, pp. 447–457, Oct. 2019.

[40] M. Xue, H. Wu, R. Li, M. Xu, and P. Jiao, "Eosdnn: An efficient offloading scheme for dnn inference acceleration in local-edge-cloud collaborative environments," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 248–264, Sep. 2021.

[41] X. Zhang, M. Mounesan, and S. Debroy, "Effect-dnn: Energy-efficient edge framework for real-time dnn inference," in *Proc. IEEE WoWMoM*, Aug. 2023, pp. 10–20.

[42] Y. Chen, J. Zhao, Y. Wu, J. Huang, and X. Shen, "Qoe-aware decentralized task offloading and resource allocation for end-edge-cloud systems: A game-theoretical approach," *IEEE Trans. Mobi. Comput.*, vol. 23, no. 1, pp. 769–784, 2024, doi=10.1109/TMC.2022.3223119.

[43] X. Tang, C. Cao, Y. Wang, S. Zhang, Y. Liu, M. Li, and T. He, "Computing power network: The architecture of convergence of computing and networking towards 6g requirement," *China commun.*, vol. 18, no. 2, pp. 175–185, Feb. 2021.

[44] Y. Sun, B. Lei, J. Liu, H. Huang, X. Zhang, J. Peng, and W. Wang, "Computing power network: A survey," *arXiv preprint arXiv:2210.06080*, Nov. 2022.

[45] Y. Cui, K. Cao, J. Zhou, and T. Wei, "Optimizing training efficiency and cost of hierarchical federated learning in heterogeneous mobile-edge cloud computing," *IEEE Trans. Computer-Aided Des. Integ. Circ. Sys.*, vol. 42, no. 5, pp. 1518–1531, May 2023.

[46] S. Zhang and D. Zhu, "Towards artificial intelligence enabled 6g: State of the art, challenges, and opportunities," *Computer Networks*, vol. 183, p. 107556, Dec. 2020.

[47] H. Djigal, J. Xu, L. Liu, and Y. Zhang, "Machine and deep learning for resource allocation in multi-access edge computing: A survey," *IEEE Commun. Surveys & Tutor.*, vol. 24, no. 4, pp. 2449–2494, Aug. 2022.

[48] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, Nov. 2023.

[49] Y. Wang, Y. Pan, M. Yan, Z. Su, and T. H. Luan, "A survey on chatgpt: Ai-generated contents, challenges, and solutions," *arXiv preprint arXiv:2305.18339*, Jul. 2023.

[50] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, Nov. 2023.

[51] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys & Tutor.*, vol. 22, no. 2, pp. 869–904, Jan. 2020.

[52] S. Duan, D. Wang, J. Ren, F. Lyu, Y. Zhang, H. Wu, and X. Shen, "Distributed artificial intelligence empowered by end-edge-cloud computing: A survey," *IEEE Commun. Surveys & Tutor.*, vol. 25, no. 1, pp. 591–624, Nov. 2022.

[53] D. Rosendo, A. Costan, P. Valduriez, and G. Antoniu, "Distributed intelligence on the edge-to-cloud continuum: A systematic literature review," *J. Parall. Distri. Comput.*, vol. 166, pp. 71–94, Aug. 2022.

[54] Z. Lin, G. Qu, Q. Chen, X. Chen, Z. Chen, and K. Huang, "Pushing large language models to the 6g edge: Vision, challenges, and opportunities," *arXiv preprint arXiv:2309.16739*, Sep. 2023.

[55] M. Xu, H. Du, D. Niyato, J. Kang, Z. Xiong, S. Mao, Z. Han, A. Jamalipour, D. I. Kim, V. Leung *et al.*, "Unleashing the power of edge-cloud generative ai in mobile networks: A survey of aigc services," *arXiv preprint arXiv:2303.16129*, Oct. 2023.

[56] H. Du, R. Zhang, D. Niyato, J. Kang, Z. Xiong, D. I. Kim, X. S. Shen, and H. V. Poor, "Exploring collaborative distributed diffusion-based ai-generated content (aigc) in wireless networks," *IEEE Network*, no. 99, pp. 1–8, Jul. 2023, doi:10.1109/MNET.006.2300223.

[57] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou *et al.*, "The rise and potential of large language model based agents: A survey," *arXiv preprint arXiv:2309.07864*, Sep. 2023.

[58] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler *et al.*, "Emergent abilities of large language models," *arXiv preprint arXiv:2206.07682*, Oct. 2022.

[59] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *J. Mach. Learn. Res. (JMLR)*, vol. 24, no. 240, pp. 1–113, Aug. 2023.

[60] C. Zhang and X. Yu, "Domestic large model technology and medical applications analysis." *Advanced Ultrasound in Diagnosis & Therapy (AUDT)*, vol. 7, no. 2, Jun. 2023.

[61] D. Guo, H. Chen, R. Wu, and Y. Wang, "Aigc challenges and opportunities related to public safety: a case study of chatgpt," *J. Saf. Sci. Resil.*, vol. 4, no. 4, pp. 329–339, Dec. 2023.

[62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *MIT press NeurIPS*, vol. 30, Dec. 2017.

[63] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang *et al.*, "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, 2021.

[64] H. Huang, O. Zheng, D. Wang, J. Yin, Z. Wang, S. Ding, H. Yin, C. Xu, R. Yang, Q. Zheng *et al.*, "Chatgpt for shaping the future of dentistry: the potential of multi-modal large language model," *International Journal of Oral Science*, vol. 15, no. 1, p. 29, Jul. 2023.

[65] Z. Tang, Z. Yang, C. Zhu, M. Zeng, and M. Bansal, "Any-to-any generation via composable diffusion," *arXiv preprint arXiv:2305.11846*, May 2023.

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

[67] C. Li, Z. Gan, Z. Yang, J. Yang, L. Li, L. Wang, and J. Gao, "Multimodal foundation models: From specialists to general-purpose assistants," *arXiv preprint arXiv:2309.10020*, vol. 1, no. 2, Sep. 2023.

[68] B. Shen, J. Zhang, T. Chen, D. Zan, B. Geng, A. Fu, M. Zeng, A. Yu, J. Ji, J. Zhao *et al.*, "Pangu-coder2: Boosting large language models for code with ranking feedback," *arXiv preprint arXiv:2307.14936*, Jul. 2023.

[69] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl *et al.*, "Large language models encode clinical knowledge," *Nature*, vol. 620, no. 7972, pp. 172–180, Jul. 2023.

[70] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "Bloomberggpt: A large language model for finance," *arXiv preprint arXiv:2303.17564*, May 2023.

[71] L. Makatura, M. Foshey, B. Wang, F. HähnLein, P. Ma, B. Deng, M. Tjandrasuwita, A. Spielberg, C. E. Owens, P. Y. Chen *et al.*, "How can large language models help humans in design and manufacturing?" *arXiv preprint arXiv:2307.14377*, Jul. 2023.

[72] G. Dong, H. Yuan, K. Lu, C. Li, M. Xue, D. Liu, W. Wang, Z. Yuan, C. Zhou, and J. Zhou, "How abilities in large language models are affected by supervised fine-tuning data composition," *arXiv preprint arXiv:2310.05492*, Nov. 2023.

[73] R. Zheng, S. Dou, S. Gao, Y. Hua, W. Shen, B. Wang, Y. Liu, S. Jin, Q. Liu, Y. Zhou *et al.*, "Secrets of rlhf in large language models part i: Ppo," *arXiv preprint arXiv:2307.04964*, Jul. 2023.

[74] K. Cao, J. Weng, and K. Li, "Reliability-driven end–end–edge collaboration for energy minimization in large-scale cyber-physical systems," *IEEE Trans. Reliability*, Aug. 2023, 10.1109/TR.2023.3297124.

[75] S. Yang, P. Yang, J. Chen, Q. Ye, N. Zhang, and X. Shen, "Delay-optimized multi-user vr streaming via end-edge collaborative neural frame interpolation," *IEEE Trans. Netw. Sci. Eng.*, Jul. 2023, 10.1109/TNSE.2023.3296511.

[76] W. Cheng, M. Zhang, F. Dong, and S. Fu, "Accelerate multi-view inference with end-edge collaborative computing," in *Proc. IEEE CSCWD*. IEEE, May 2023, pp. 1625–1631.

[77] J. Zhang, Z. Qu, C. Chen, H. Wang, Y. Zhan, B. Ye, and S. Guo, "Edge learning: The enabling technology for distributed big data analytics in the edge," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–36, Jul. 2021.

[78] I. Behnke and H. Austad, "Real-time performance of industrial iot communication technologies: A review," *IEEE Internet of Things J.*, Nov. 2023, 10.1109/JIOT.2023.3332507.

[79] J. Qi, H. Zhang, X. Li, H. Ji, and X. Shao, "Edge-edge collaboration based micro-service deployment in edge computing networks," in *Proc. IEEE WCNC*, May. 2023, pp. 1–6.

[80] Y. Xu, L. Chen, Z. Lu, X. Du, J. Wu, and P. C. K. Hung, "An adaptive mechanism for dynamically collaborative computing power and task scheduling in edge environment," *IEEE Internet of Things J.*, vol. 10, no. 4, pp. 3118–3129, Oct. 2023.

[81] Y. Nan, S. Jiang, and M. Li, "Large-scale video analytics with cloud–edge collaborative continuous learning," *ACM Tran. Sen. Netw.*, vol. 20, no. 1, pp. 1–23, Oct. 2023.

[82] J. Yao, S. Zhang, Y. Yao, F. Wang, J. Ma, J. Zhang, Y. Chu, L. Ji, K. Jia, T. Shen *et al.*, "Edge-cloud polarization and collaboration: A comprehensive survey for ai," *IEEE Trans. Know. Data Eng.*, vol. 35, no. 7, pp. 6866–6886, Jul. 2023.

[83] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–36, Oct. 2019.

[84] Y. Chen, R. Li, Z. Zhao, C. Peng, J. Wu, E. Hossain, and H. Zhang, "Netgpt: A native-ai network architecture beyond provisioning personalized generative services," *arXiv preprint arXiv:2307.06148*, Jul. 2023.

[85] M. Teng, X. Li, and K. Zhu, "Joint optimization of sequential task offloading and service deployment in end-edge-cloud system for energy efficiency," *IEEE Trans. Sustain. Comput.*, 2023, 10.1109/TSUSC.2023.3291365.

[86] G. Liu, F. Dai, X. Xu, X. Fu, W. Dou, N. Kumar, and M. Bilal, "An adaptive dnn inference acceleration framework with end–edge–cloud collaborative computing," *Future Generation Computer Systems*, vol. 140, pp. 422–435, Mar. 2023.

[87] S. Shahhosseini, D. Seo, A. Kanduri, T. Hu, S.-S. Lim, B. Donyanavard, A. M. Rahmani, and N. Dutt, "Online learning for orchestration

of inference in multi-user end-edge-cloud networks," *ACM Trans. Embedded Comput. Sys.*, vol. 21, no. 6, pp. 1–25, Dec. 2022.

[88] L. Huawei Technologies Co., "Huawei atlas ai computing solution," in *Artificial Intelligence Technology*. Springer, Oct. 2022, pp. 163–219.

[89] D. S. Carrión and V. Prohaska, "Exploration of tpus for ai applications," *arXiv preprint arXiv:2309.08918*, Nov. 2023.

[90] R. Muralidhar, R. Borovica-Gajic, and R. Buyya, "Energy efficient computing systems: Architectures, abstractions and modeling to techniques and standards," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, Sep. 2022.

[91] Y.-H. Lai, E. Ustun, S. Xiang, Z. Fang, H. Rong, and Z. Zhang, "Programming and synthesis for software-defined fpga acceleration: status and future prospects," *ACM Trans. Reconf. Tech. Sys.*, vol. 14, no. 4, pp. 1–39, Sep. 2021.

[92] P. Maillard, Y. P. Chen, J. Vidmar, N. Fraser, G. Gambardella, M. Sawant, and M. L. Voogel, "Radiation-tolerant deep learning processor unit (dpu)-based platform using xilinx 20-nm kintex ultrascale fpga," *IEEE Trans. Nucl. Sci.*, vol. 70, no. 4, pp. 714–721, Oct. 2022.

[93] K. Alizadeh, I. Mirzadeh, D. Belenko, K. Khatamifard, M. Cho, C. C. Del Mundo, M. Rastegari, and M. Farajtabar, "Llm in a flash: Efficient large language model inference with limited memory," *arXiv preprint arXiv:2312.11514*, Jan. 2024.

[94] S. Tarkoma, R. Morabito, and J. Sauvola, "Ai-native interconnect framework for integration of large language model technologies in 6g systems," *arXiv preprint arXiv:2311.05842*, Nov. 2023.

[95] X. Ma, G. Fang, and X. Wang, "Llm-pruner: On the structural pruning of large language models," *arXiv preprint arXiv:2305.11627*, Sep. 2023.

[96] M. Sun, Z. Liu, A. Bair, and J. Z. Kolter, "A simple and effective pruning approach for large language models," *arXiv preprint arXiv:2306.11695*, Oct. 2023.

[97] J. Liu, R. Gong, X. Wei, Z. Dong, J. Cai, and B. Zhuang, "Qllm: Accurate and efficient low-bitwidth quantization for large language models," *arXiv preprint arXiv:2310.08041*, Oct. 2023.

[98] C.-Y. Hsieh, C.-L. Li, C.-K. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister, "Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes," *arXiv preprint arXiv:2305.02301*, Jul. 2023.

[99] N. Zhou, H. Zhou, and D. Hoppe, "Containerization for high performance computing systems: Survey and prospects," *IEEE Trans. Soft. Eng.*, vol. 49, no. 4, pp. 2722–2740, Dec. 2022.

[100] C. Carrión, "Kubernetes scheduling: Taxonomy, ongoing issues and challenges," *ACM Computing Surveys (CSUR)*, vol. 55, no. 7, pp. 1–37, Dec. 2022.

[101] A. Jeffery, H. Howard, and R. Mortier, "Rearchitecting kubernetes for the edge," in *Proc. ACM EdgeSys*, Apr. 2021, pp. 7–12.

[102] T. Nguyen, C. Van Nguyen, V. D. Lai, H. Man, N. T. Ngo, F. Dernoncourt, R. A. Rossi, and T. H. Nguyen, "Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages," *arXiv preprint arXiv:2309.09400*, Sep. 2023.

[103] X. Li, S. Guo, and P. Li, "Energy-efficient data collection scheme based on mobile edge computing in wsns," in *Proc. IEEE MSN*, Dec. 2019, pp. 95–100.

[104] R. Anwit, P. K. Jana, and A. Tomar, "Sustainable and optimized data collection via mobile edge computing for disjoint wireless sensor networks," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 2, pp. 471–484, Apr. 2022.

[105] X. Zheng, L. Tian, B. Hui, and X. Liu, "Distributed and privacy preserving graph data collection in internet of thing systems," *IEEE Internet of Things J.*, vol. 9, no. 12, pp. 9301–9309, Jun. 2021.

[106] M. Usman, M. A. Jan, A. Jolfaei, M. Xu, X. He, and J. Chen, "A distributed and anonymous data collection framework based on multilevel edge computing architecture," *IEEE Trans. Indu. Inform.*, vol. 16, no. 9, pp. 6114–6123, Nov. 2019.

[107] L. Liu, Z. Lu, L. Wang, Y. Chen, X. Wen, Y. Liu, and M. Li, "Evenness-aware data collection for edge-assisted mobile crowdsensing in internet of vehicles," *IEEE Internet of Things J.*, vol. 10, no. 1, pp. 1–16, Jan. 2023.

[108] B. Liu, H. Zhang, J. Liu, and Q. Wang, "Acigs: An automated large-scale crops image generation system based on large visual language multi-modal models," in *Proc. IEEE SECON*, Sep. 2023, pp. 7–13.

[109] A. Tawakuli, D. Kaiser, and T. Engel, "Transforming iot data preprocessing: A holistic, normalized and distributed approach," in *Proc. ACM SenSys*, Nov. 2022, pp. 1083–1088.

[110] T. Wang, H. Ke, X. Zheng, K. Wang, A. K. Sangaiah, and A. Liu, "Big data cleaning based on mobile edge computing in industrial sensor-cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1321–1329, Sep. 2020.

[111] H. Gu, L. Zhao, Z. Han, G. Zheng, and S. Song, "Ai-enhanced cloud-edge-terminal collaborative network: Survey, applications, and future directions," *IEEE Communications Surveys & Tutorials*, Dec. 2023, doi=10.1109/COMST.2023.3338153.

[112] T. Che, J. Liu, Y. Zhou, J. Ren, J. Zhou, V. S. Sheng, H. Dai, and D. Dou, "Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization," *arXiv preprint arXiv:2310.15080*, Oct. 2023.

[113] W. Zhuang, C. Chen, and L. Lyu, "When foundation model meets federated learning: Motivations, challenges, and future directions," *arXiv preprint arXiv:2306.15546*, Jun. 2023.

[114] S. Yu, J. P. Muñoz, and A. Jannesari, "Federated foundation models: Privacy-preserving and collaborative learning for large models," *arXiv preprint arXiv:2305.11414*, Nov. 2023.

[115] W. Kuang, B. Qian, Z. Li, D. Chen, D. Gao, X. Pan, Y. Xie, Y. Li, B. Ding, and J. Zhou, "Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning," *arXiv preprint arXiv:2309.00363*, Sep. 2023.

[116] J. Jiang, X. Liu, and C. Fan, "Low-parameter federated learning with large language models," *arXiv preprint arXiv:2307.13896*, Jul. 2023.

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

[118] H. Li, K. D. R. Assis, S. Yan, and D. Simeonidou, "Drl-based long-term resource planning for task offloading policies in multiserver edge computing networks," *IEEE Trans. Netw. Serv. Mana.*, vol. 19, no. 4, pp. 4151–4164, 2022.

[119] Z. Tang, Y. Wang, X. He, L. Zhang, X. Pan, Q. Wang, R. Zeng, K. Zhao, S. Shi, B. He *et al.*, "Fusionai: Decentralized training and deploying llms with massive consumer-level gpus," *arXiv preprint arXiv:2309.01172*, Sep. 2023.

[120] Z. Di, T. Luo, C. Qiu, C. Zhang, Z. Liu, X. Wang, and J. Jiang, "In-network pooling: Contribution-aware allocation optimization for computing power network in b5g/6g era," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 3, pp. 1190–1202, May 2023.

[121] J. Liu, K. Luo, Z. Zhou, and X. Chen, "Erp: Edge resource pooling for data stream mobile computing," *IEEE Internet of Things J.*, vol. 6, no. 3, pp. 4355–4368, Nov. 2019.

[122] F. Lyu, J. Ren, P. Yang, N. Cheng, Y. Zhang, and X. Shen, "Sosa: Socializing static aps for edge resource pooling in large-scale wifi system," in *Proc. IEEE INFOCOM*, Aug. 2020, pp. 1181–1190.

[123] Y. Sahni, J. Cao, L. Yang, and S. Wang, "Distributed resource scheduling in edge computing: Problems, solutions, and opportunities," *Computer Networks*, vol. 219, p. 109430, Dec. 2022.

[124] J. Li, H. Lv, B. Lei, and Y. Xie, "Modeling and optimization for computing power resource-aware in cpn," in *Proc. IEEE APNOMS*, Sep. 2023, pp. 353–356.

[125] S. R. Jeremiah, L. T. Yang, and J. H. Park, "Digital twin-assisted resource allocation framework based on edge collaboration for vehicular edge computing," *Future Generation Computer Systems*, vol. 150, pp. 243–254, Jan. 2024.

[126] J. Ge, K. Wu, N. Jamal, and F. Ullah, "Dynamic resource allocation techniques for wireless network data in elastic optical network applications," *Mobile Networks and Applications*, pp. 1–12, Oct. 2023.

[127] M. Murakami, T. Kurimoto, S. Okamoto, and N. Yamanaka, "Experimental evaluation on priority-aware guaranteed resource allocation for resource pool based reconfigurable hardware," *IEEE/ACM Trans. Netw.*, pp. 1–10, Jun. 2023.

[128] T. Yeh and S. Yu, "Realizing dynamic resource orchestration on cloud systems in the cloud-to-edge continuum," *J. Para. and Dist. Comput.*, vol. 160, pp. 100–109, Feb. 2022.

[129] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing," *IEEE Trans. Intel. Trans. Sys.*, Mar. 2023, doi:10.1109/TITS.2023.3249745.

[130] W.-C. Chien, C.-F. Lai, and H.-C. Chao, "Dynamic resource prediction and allocation in c-ran with edge artificial intelligence," *IEEE Trans. Indust. Inform.*, vol. 15, no. 7, pp. 4306–4314, Apr. 2019.

[131] F. Tusa and S. Clayman, "End-to-end slices to orchestrate resources and services in the cloud-to-edge continuum," *Futu. Gener. Comput. Sys.*, vol. 141, pp. 473–488, Apr. 2023.

[132] X. Wang, X. Ren, C. Qiu, Z. Xiong, H. Yao, and V. C. M. Leung, "Integrating edge intelligence and blockchain: What, why, and how," *IEEE Commun. Surveys & Tutor.*, vol. 24, no. 4, pp. 2193–2229, Jul. 2022.

[133] Y. Li, M. Xia, J. Duan, and Y. Chen, "Pricing-based resource allocation in three-tier edge computing for social welfare maximization," *Comput. Netw.*, vol. 217, p. 109311, Nov. 2022.

[134] Y. Chen, J. Zhao, J. Hu, S. Wan, and J. Huang, "Distributed task offloading and resource purchasing in noma-enabled mobile edge computing: Hierarchical game theoretical approaches," *ACM Trans. Embedded Comput. Sys.*, 2023, doi:https://doi.org/10.1145/3597023.

[135] T. Faisal, J. A. O. Lucena, D. R. Lopez, C. Wang, and M. Dohler, "How to design autonomous service level agreements for 6g," *IEEE Commun. Magazine*, vol. 61, no. 3, pp. 80–85, Mar. 2023.

[136] Y. Du, Z. Wang, J. Li, L. Shi, D. N. K. Jayakody, Q. Chen, W. Chen, and Z. Han, "Blockchain-aided edge computing market: Smart contract and consensus mechanisms," *IEEE Trans. Mobi. Comput.*, vol. 22, no. 6, pp. 3193–3208, Jun. 2023.

[137] T. Kwantwi, G. Sun, N. A. E. Kuadey, G. Maale, and G. Liu, "Blockchain-based computing resource trading in autonomous multi-access edge network slicing: A dueling double deep q-learning approach," *IEEE Trans. Netw. Ser. Manage.*, pp. 1–1, 2023, doi:10.1109/TNSM.2023.3240301.

[138] H. Zhang, S. Huang, M. Xu, D. Guo, X. Wang, V. C. Leung, and W. Wang, "How far have edge clouds gone? a spatial-temporal analysis of edge network latency in the wild," in *Proc. IEEE/ACM IWQoS*. IEEE, 2023, pp. 1–10.

[139] X. Ren, C. Qiu, X. Wang, Z. Han, K. Xu, H. Yao, and S. Guo, "Ai-bazaar: A cloud-edge computing power trading framework for ubiquitous ai services," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 2337–2348, Sep. 2023.

[140] D. Liu, X. Chen, Z. Zhou, and Q. Ling, "Hiertrain: Fast hierarchical edge ai learning with hybrid parallelism in mobile-edge-cloud computing," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 634–645, May 2020.

[141] Y. Li, Z. Zeng, J. Li, B. Yan, Y. Zhao, and J. Zhang, "Distributed model training based on data parallelism in edge computing-enabled elastic optical networks," *IEEE Communications Letters*, vol. 25, no. 4, pp. 1241–1244, Apr. 2021.

[142] S. Fan, Y. Rong, C. Meng, Z. Cao, S. Wang, Z. Zheng, C. Wu, G. Long, J. Yang, L. Xia *et al.*, "Dapple: A pipelined data parallel approach for training large models," in *Proc. ACM SIGPLAN Symposium Prin. Prac. Para. Prog.*, Feb. 2021, pp. 431–445.

[143] T. Sen and H. Shen, "A data and model parallelism based distributed deep learning system in a network of edge devices," in *Proc. IEEE ICCCN*, Sep. 2023, pp. 1–10.

[144] H. Zhou, M. Li, N. Wang, G. Min, and J. Wu, "Accelerating deep learning inference via model parallelism and partial computation offloading," *IEEE Trans. Para. Distri. Sys.*, vol. 34, no. 2, pp. 475–488, Feb. 2023.

[145] P. Li, E. Koyuncu, and H. Seferoglu, "Adaptive and resilient model-distributed inference in edge computing systems," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 1263–1273, May. 2023.

[146] Z. Li, L. Zheng, Y. Zhong, V. Liu, Y. Sheng, X. Jin, Y. Huang, Z. Chen, H. Zhang, J. E. Gonzalez *et al.*, "{AlpaServe}: Statistical multiplexing with model parallelism for deep learning serving," in *USENIX Symposium OSDI*, 2023, pp. 663–679.

[147] P. Li, E. Koyuncu, and H. Seferoglu, "Respipe: Resilient model-distributed dnn training at edge networks," in *Proc. IEEE ICASSP*, May 2021, pp. 3660–3664.

[148] Y. Chen, Q. Yang, S. He, Z. Shi, J. Chen, and M. Guizani, "Ftpipehd: A fault-tolerant pipeline-parallel distributed training approach for heterogeneous edge devices," *IEEE Trans. Mobi. Comput.*, pp. 1–13, Jun. 2023, doi=10.1109/TMC.2023.3272567.

[149] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Are. in Commun.*, vol. 37, no. 6, pp. 1205–1221, Mar. 2019.

[150] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE ICC*, Jul. 2020, pp. 1–6.

[151] J. Ren, G. Yu, and G. Ding, "Accelerating dnn training in wireless federated edge learning systems," *IEEE J. Sel. Are. in Commun.*, vol. 39, no. 1, pp. 219–232, Jan. 2021.

[152] E. Tanghatari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Distributing dnn training over iot edge devices based on transfer learning," *Neurocomputing*, vol. 467, pp. 56–65, Jan. 2022.

[153] X. He, Q. Chen, L. Tang, W. Wang, T. Liu, L. Li, Q. Liu, and L. jia., "Federated continuous learning based on stacked broad learning system assisted by digital twin networks: An incremental learning approach for intrusion detection in uav networks," *IEEE Internet of Things J.*, pp. 1–1, Jun. 2023, doi=10.1109/JIOT.2023.3282648.

[154] M. A. Hussain, S.-A. Huang, and T.-H. Tsai, "Learning with sharing: An edge-optimized incremental learning method for deep neural networks," *IEEE Trans. Emerg. Topics in Comput.*, vol. 11, no. 2, pp. 461–473, Oct. 2022.

[155] F. Bu, C. Hu, Q. Zhang, C. Bai, L. T. Yang, and T. Baker, "A cloud-edge-aided incremental high-order possibilistic c-means algorithm for medical data clustering," *IEEE Trans. Fuzzy Sys.*, vol. 29, no. 1, pp. 148–155, Jan. 2021.

[156] X. Liu, M. Derakhshani, and S. Lambotharan, "Contextual learning for content caching with unknown time-varying popularity profiles via incremental clustering," *IEEE Trans. Commun.*, vol. 69, no. 5, pp. 3011–3024, Feb. 2021.

[157] C. Gong, F. Lin, X. Gong, and Y. Lu, "Intelligent cooperative edge computing in internet of things," *IEEE Internet of Things J.*, vol. 7, no. 10, pp. 9372–9382, Apr. 2020.

[158] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "Smoothquant: Accurate and efficient post-training quantization for large language models," in *Proc. ACM ICML*, xxx. 2023, pp. 38 087–38 099.

[159] A. Banitalebi-Dehkordi, N. Vedula, J. Pei, F. Xia, L. Wang, and Y. Zhang, "Auto-split: A general framework of collaborative edge-cloud ai," in *Proc. ACM SIGKDD*, Aug. 2021, pp. 2543–2553.

[160] Q. Zhou, S. Guo, Z. Qu, J. Guo, Z. Xu, J. Zhang, T. Guo, B. Luo, and J. Zhou, "Octo:{INT8} training with loss-aware compensation and backward quantization for tiny on-device learning," in *Proc. USENIX ATC*, 2021, pp. 177–191.

[161] S. Roy, "Understanding the impact of post-training quantization on large-scale language models," *arXiv preprint arXiv:2309.05210*, Sep. 2023.

[162] X. Zhu, J. Li, Y. Liu, C. Ma, and W. Wang, "A survey on model compression for large language models," *arXiv preprint arXiv:2308.07633*, Aug. 2023.

[163] H. Wang, C. Imes, S. Kundu, P. A. Beerel, S. P. Crago, and J. P. Walters, "Quantpipe: Applying adaptive post-training quantization for distributed transformer pipelines in dynamic edge environments," in *Proc. IEEE ICASSP*, May. 2023, pp. 1–5.

[164] S. J. Kwon, J. Kim, J. Bae, K. M. Yoo, J.-H. Kim, B. Park, B. Kim, J.-W. Ha, N. Sung, and D. Lee, "Alphatuning: Quantization-aware parameter-efficient adaptation of large-scale pre-trained language models," *arXiv preprint arXiv:2210.03858*, May 2022.

[165] Q. Li, Y. Zhang, L. Li, P. Yao, B. Zhang, X. Chu, Y. Sun, L. Du, and Y. Xie, "Fptq: Fine-grained post-training quantization for large language models," *arXiv preprint arXiv:2308.15987*, Aug. 2023.

[166] M. Kirtas, A. Oikonomou, N. Passalis, G. Mourgias-Alexandris, M. Moralis-Pegios, N. Pleros, and A. Tefas, "Quantization-aware training for low precision photonic neural networks," *Neural Networks*, vol. 155, pp. 561–573, Nov. 2022.

[167] Z. Liu, B. Oguz, C. Zhao, E. Chang, P. Stock, Y. Mehdad, Y. Shi, R. Krishnamoorthi, and V. Chandra, "Llm-qat: Data-free quantization aware training for large language models," *arXiv preprint arXiv:2305.17888*, May 2023.

[168] C. Yu, T. Chen, and Z. Gan, "Boost transformer-based language models with gpu-friendly sparsity and quantization," in *Findings of the Association for Computational Linguistics: ACL 2023*, Jul. 2023, pp. 218–235.

[169] X. Xia, H. Yin, J. Yu, Q. Wang, G. Xu, and Q. V. H. Nguyen, "On-device next-item recommendation with self-supervised knowledge distillation," in *Proc. ACM SIGIR*, Aug. 2022, pp. 546–555.

[170] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Trans. Patt. Analy. Mach. Intell.*, vol. 44, no. 6, pp. 3048–3068, Jan. 2021.

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

[172] R. He, S. Sun, J. Yang, S. Bai, and X. Qi, "Knowledge distillation as efficient pre-training: Faster convergence, higher data-efficiency, and better transferability," in *Proc. IEEE/CVF CVPR*, 2022, pp. 9161–9171.

[173] C. Liang, S. Zuo, Q. Zhang, P. He, W. Chen, and T. Zhao, "Less is more: Task-aware layer-wise distillation for language model compression," in *Proc. ACM ICML*, xxx. 2023, pp. 20 852–20 867.

[174] S. Javed, A. Mahmood, T. Qaiser, and N. Werghi, "Knowledge distillation in histology landscape by multi-layer features supervision," *IEEE J. Bio. Heal. Inform.*, vol. 27, no. 4, pp. 2037–2046, Jan. 2023.

[175] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, Sep. 2019.

[176] M. Zhang, N. U. Naresh, and Y. He, "Adversarial data augmentation for task-specific knowledge distillation of pre-trained transformers," in *Proc. AAAI*, vol. 36, no. 10, Jun. 2022, pp. 11 685–11 693.

[177] M. Kim, S. Lee, S. Hong, D.-S. Chang, and J. Choi, "Understanding and improving knowledge distillation for quantization-aware training of large transformer encoders," *arXiv preprint arXiv:2211.11014*, Nov. 2022.

[178] Y.-C. Wang, J. Xue, C. Wei, and C.-C. J. Kuo, "An overview on generative ai at scale with edge-cloud computing," *IEEE Open J. Commun. Society*, 2023, doi:10.1109/OJCOMS.2022.1234567.

[179] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," *arXiv preprint arXiv:2306.02858*, Jun. 2023.

[180] N. Hegde, S. Paul, G. Madan, and G. Aggarwal, "Analyzing the efficacy of an llm-only approach for image-based document question answering," *arXiv preprint arXiv:2309.14389*, Sep. 2023.

[181] S. Moon, A. Madotto, Z. Lin, T. Nagarajan, M. Smith, S. Jain, C.-F. Yeh, P. Murugesan, P. Heidari, Y. Liu *et al.*, "Anymal: An efficient and scalable any-modality augmented language model," *arXiv preprint arXiv:2309.16058*, Sep. 2023.

[182] J. Y. Koh, D. Fried, and R. Salakhutdinov, "Generating images with multimodal language models," *arXiv preprint arXiv:2305.17216*, May 2023.

[183] L. Lian, B. Li, A. Yala, and T. Darrell, "Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models," *arXiv preprint arXiv:2305.13655*, Oct. 2023.

[184] S. Baek, H. Im, J. Ryu, J. Park, and T. Lee, "Promptcrafter: Crafting text-to-image prompt through mixed-initiative dialogue with llm," *arXiv preprint arXiv:2307.08985*, Jul. 2023.

[185] C. Qin, S. Zhang, N. Yu, Y. Feng, X. Yang, Y. Zhou, H. Wang, J. C. Niebles, C. Xiong, S. Savarese *et al.*, "Unicontrol: A unified diffusion model for controllable visual generation in the wild," *arXiv preprint arXiv:2305.11147*, May 2023.

[186] H. Gani, S. F. Bhat, M. Naseer, S. Khan, and P. Wonka, "Llm blueprint: Enabling text-to-image generation with complex and detailed prompts," *arXiv preprint arXiv:2310.10640*, Oct. 2023.

[187] H. Lin, A. Zala, J. Cho, and M. Bansal, "Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning," *arXiv preprint arXiv:2309.15091*, Sep. 2023.

[188] J. Gu, S. Wang, H. Zhao, T. Lu, X. Zhang, Z. Wu, S. Xu, W. Zhang, Y.-G. Jiang, and H. Xu, "Reuse and diffuse: Iterative denoising for text-to-video generation," *arXiv preprint arXiv:2309.03549*, Sep. 2023.

[189] S. Hong, J. Seo, S. Hong, H. Shin, and S. Kim, "Large language models are frame-level directors for zero-shot text-to-video generation," *arXiv preprint arXiv:2305.14330*, May 2023.

[190] L. Shen, Y. Zhang, H. Zhang, and Y. Wang, "Data player: Automatic generation of data videos with narration-animation interplay," *arXiv preprint arXiv:2308.04703*, Aug. 2023.

[191] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua, "Next-gpt: Any-to-any multimodal llm," *arXiv preprint arXiv:2309.05519*, Sep. 2023.

[192] K. Zheng, X. He, and X. E. Wang, "Minigpt-5: Interleaved vision-and-language generation via generative vokens," *arXiv preprint arXiv:2310.02239*, Oct. 2023.

[193] L. Wang, Y. Ling, Z. Yuan, M. Shridhar, C. Bao, Y. Qin, B. Wang, H. Xu, and X. Wang, "Gensim: Generating robotic simulation tasks via large language models," *arXiv preprint arXiv:2310.01361*, Oct. 2023.

[194] Z. Zhong, D. Rempe, Y. Chen, B. Ivanovic, Y. Cao, D. Xu, M. Pavone, and B. Ray, "Language-guided traffic simulation via scene-level diffusion," *arXiv preprint arXiv:2306.06344*, Jun. 2023.

[195] Y. Lin, H. Wu, R. Wang, H. Lu, X. Lin, H. Xiong, and L. Wang, "Towards language-guided interactive 3d generation: Llms as layout interpreter with generative feedback," *arXiv preprint arXiv:2305.15808*, May 2023.

[196] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan, "3d-llm: Injecting the 3d world into large language models," *arXiv preprint arXiv:2307.12981*, Jul. 2023.

[197] N. Nascimento, P. Alencar, and D. Cowan, "Gpt-in-the-loop: Adaptive decision-making for multiagent systems," *arXiv preprint arXiv:2308.10435*, 2023.

[198] H. Zou, Q. Zhao, L. Bariah, M. Bennis, and M. Debbah, "Wireless multi-agent generative ai: From connected intelligence to collective intelligence," *arXiv preprint arXiv:2307.02757*, Jul. 2023.

[199] L. Chen, Y. Zhang, S. Ren, H. Zhao, Z. Cai, Y. Wang, T. Liu, and B. Chang, "Towards end-to-end embodied decision making via multi-modal large language model: Explorations with gpt4-vision and beyond," *arXiv preprint arXiv:2310.02071*, Oct. 2023.

[200] Y. Xie, T. Xie, M. Lin, W. Wei, C. Li, B. Kong, L. Chen, C. Zhuo, B. Hu, and Z. Li, "Olagpt: Empowering llms with human-like problem-solving abilities," *arXiv preprint arXiv:2305.16334*, May 2023.

[201] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," in *Proc. IEEE/CVF ICCV*, 2023, pp. 2998–3009.

[202] L. Meng, M. Wen, C. Le, X. Li, D. Xing, W. Zhang, Y. Wen, H. Zhang, J. Wang, Y. Yang *et al.*, "Offline pre-trained multi-agent decision transformer," *Machine Intelligence Research*, vol. 20, no. 2, pp. 233–248, Mar. 2023.

[203] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, May 2022.

[204] Y. Wen, Z. Wan, M. Zhou, S. Hou, Z. Cao, C. Le, J. Chen, Z. Tian, W. Zhang, and J. Wang, "On realization of intelligent decision-making in the real world: A foundation decision model perspective," *arXiv preprint arXiv:2212.12669*, Dec. 2022.

[205] Y. Yan, C. Niu, R. Gu, F. Wu, S. Tang, L. Hua, C. Lyu, and G. Chen, "On-device learning for model personalization with large-scale cloud-coordinated domain adaption," in *Proc. ACM SIGKDD*, Aug. 2022, pp. 2180–2190.

[206] J. Chen, Z. Liu, X. Huang, C. Wu, Q. Liu, G. Jiang, Y. Pu, Y. Lei, X. Chen, X. Wang *et al.*, "When large language models meet personalization: Perspectives of challenges and opportunities," *arXiv preprint arXiv:2307.16376*, Jun. 2023.

[207] D. Huang, Z. Wei, A. Yue, X. Zhao, Z. Chen, R. Li, K. Jiang, B. Chang, Q. Zhang, S. Zhang *et al.*, "Dsqa-llm: Domain-specific intelligent question answering based on large language model," in *International Conference on AI-generated Content*. Springer, Nov. 2023, pp. 170–180.

[208] P. Fu, Y. Zhang, H. Wang, W. Qiu, and J. Zhao, "Revisiting the knowledge injection frameworks," *arXiv preprint arXiv:2311.01150*, Nov. 2023.

[209] A. Agarwal, S. Gawade, A. P. Azad, and P. Bhattacharyya, "Kitlm: Domain-specific knowledge integration into language models for question answering," *arXiv preprint arXiv:2308.03638*, Aug. 2023.

[210] S. Dingliwa, A. Shenoy, S. Bodapati, A. Gandhe, R. T. Gadde, and K. Kirchhoff, "Domain prompts: Towards memory and compute efficient domain adaptation of asr systems," in *Interspeech 2022*, 2022. [Online]. Available: https://www.amazon.science/publications/domain-prompts-towards-memory-and-compute-efficient-domain-adaptation-of-asr-systems

[211] Y. Wen, Z. Wang, and J. Sun, "Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models," *arXiv preprint arXiv:2308.09729*, Aug. 2023.

[212] G. Vernikos, A. Bražinskas, J. Adamek, J. Mallinson, A. Severyn, and E. Malmi, "Small language models improve giants by rewriting their outputs," *arXiv preprint arXiv:2305.13514*, May 2023.

[213] W. Zhang, H. Liu, Y. Du, C. Zhu, Y. Song, H. Zhu, and Z. Wu, "Bridging the information gap between domain-specific model and general llm for personalized recommendation," *arXiv preprint arXiv:2311.03778*, Nov 2023.

[214] K. Zhang, F. Zhao, Y. Kang, and X. Liu, "Memory-augmented llm personalization with short-and long-term memory coordination," *arXiv preprint arXiv:2309.11696*, Sep. 2023.

[215] C. Sun, X. Li, J. Wen, X. Wang, Z. Han, and V. C. Leung, "Federated deep reinforcement learning for recommendation-enabled edge caching in mobile edge-cloud computing networks," *IEEE J. Sel. Are. in Commun.*, vol. 41, no. 3, pp. 690–705, Mar. 2023.

[216] Z. Liu, Q. Z. Sheng, Z. Zhang, X. Xu, D. Chu, J. Yu, and S. Wang, "Accurate and reliable service recommendation based on bilateral perception in multi-access edge computing," *IEEE Trans. Serv. Comput.*, vol. 16, no. 2, pp. 886–899, Mar. 2023.

[217] L. Li, Y. Zhang, D. Liu, and L. Chen, "Large language models for generative recommendation: A survey and visionary discussions," *arXiv preprint arXiv:2309.01157*, Sep. 2023.

[218] W. Fan, Z. Zhao, J. Li, Y. Liu, X. Mei, Y. Wang, J. Tang, and Q. Li, "Recommender systems in the era of large language models (llms)," *arXiv preprint arXiv:2307.02046*, Aug. 2023.

[219] H. Lyu, S. Jiang, H. Zeng, Y. Xia, and J. Luo, "Llm-rec: Personalized recommendation via prompting large language models," *arXiv preprint arXiv:2307.15780*, Aug. 2023.

[220] X. Huang, J. Lian, Y. Lei, J. Yao, D. Lian, and X. Xie, "Recommender ai agent: Integrating large language models for interactive recommendations," *arXiv preprint arXiv:2308.16505*, Aug. 2023.

[221] B. Yin, J. Xie, Y. Qin, Z. Ding, Z. Feng, X. Li, and W. Lin, "Heterogeneous knowledge fusion: A novel approach for personalized recommendation via llm," in *Proceedings ACM Conference on Recommender Systems*, Sep. 2023, pp. 599–601.

[222] B. Yang, L. He, N. Ling, Z. Yan, G. Xing, X. Shuai, X. Ren, and X. Jiang, "Edgefm: Leveraging foundation model for open-set learning on the edge," in *Proc. ACM SenSys*, xxx 2023, pp. 1–14.

[223] H. Sha, Y. Mu, Y. Jiang, L. Chen, C. Xu, P. Luo, S. E. Li, M. Tomizuka, W. Zhan, and M. Ding, "Languagempc: Large language models as decision makers for autonomous driving," *arXiv preprint arXiv:2310.03026*, Oct. 2023.

[224] L. Wen, D. Fu, X. Li, X. Cai, T. Ma, P. Cai, M. Dou, B. Shi, L. He, and Y. Qiao, "Dilu: A knowledge-driven approach to autonomous driving with large language models," *arXiv preprint arXiv:2309.16292*, Sep. 2023.

[225] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," *arXiv preprint arXiv:2310.01412*, Oct. 2023.

[226] A. Keysan, A. Look, E. Kosman, G. Gürsun, J. Wagner, Y. Yu, and B. Rakitsch, "Can you text what is happening? integrating pre-trained language encoders into trajectory prediction models for autonomous driving," *arXiv preprint arXiv:2309.05282*, Sep. 2023.

[227] J. Mao, Y. Qian, H. Zhao, and Y. Wang, "Gpt-driver: Learning to drive with gpt," *arXiv preprint arXiv:2310.01415*, Mao 2023.

[228] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang *et al.*, "Planning-oriented autonomous driving," in *Proc. IEEE/CVF CVPR*, Jun. 2023, pp. 17 853–17 862.

[229] C. Cui, Y. Ma, X. Cao, W. Ye, and Z. Wang, "Drive as you speak: Enabling human-like interaction with large language models in autonomous vehicles," *arXiv preprint arXiv:2309.10228*, Sep. 2023.

[230] Y. Tian, X. Li, H. Zhang, C. Zhao, B. Li, X. Wang, and F.-Y. Wang, "Vistagpt: Generative parallel transformers for vehicles with intelligent systems for transport automation," *IEEE Transactions on Intelligent Vehicles*, Aug. 2023, doi:10.1109/TIV.2023.3307012.

[231] X. Li, L. Zhang, Z. Wu, Z. Liu, L. Zhao, Y. Yuan, J. Liu, G. Li, D. Zhu, P. Yan *et al.*, "Artificial general intelligence for medical imaging," *arXiv preprint arXiv:2306.05480*, Jun. 2023.

[232] M. D. Danu, G. Marica, S. K. Karn, B. Georgescu, A. Mansoor, F. Ghesu, L. M. Itu, C. Suciu, S. Grbic, O. Farri *et al.*, "Generation of radiology findings in chest x-ray by leveraging collaborative knowledge," *arXiv preprint arXiv:2306.10448*, Jun. 2023.

[233] Y. Li, J. Li, J. He, and C. Tao, "Ae-gpt: Using large language models to extract adverse events from surveillance reports-a use case with influenza vaccine adverse events," *arXiv preprint arXiv:2309.16150*, Sep. 2023.

[234] C. Dou, Z. Jin, W. Jiao, H. Zhao, Z. Tao, and Y. Zhao, "Plug-and-play medical dialogue system," *arXiv preprint arXiv:2305.11508*, May 2023.

[235] Y. Shi, S. Xu, Z. Liu, T. Liu, X. Li, and N. Liu, "Mededit: Model editing for medical question answering with external knowledge bases," *arXiv preprint arXiv:2309.16035*, Sep. 2023.

[236] J. A. Omiye, H. Gui, S. J. Rezaei, J. Zou, and R. Daneshjou, "Large language models in medicine: the potentials and pitfalls," *arXiv preprint arXiv:2309.00087*, Aug. 2023.

[237] S. Kernan Freire, M. Foosherian, C. Wang, and E. Niforatos, "Harnessing large language models for cognitive assistants in factories," in *Proc. ACM CUI*, Jul. 2023, pp. 1–6.

[238] J. Mai, J. Chen, B. Li, G. Qian, M. Elhoseiny, and B. Ghanem, "Llm as a robotic brain: Unifying egocentric memory and control," *arXiv preprint arXiv:2304.09349*, Jun. 2023.

[239] H. Liu, Y. Zhu, K. Kato, I. Kondo, T. Aoyama, and Y. Hasegawa, "Llm-based human-robot collaboration framework for manipulation tasks," *arXiv preprint arXiv:2308.14972*, Aug 2023.

[240] B. Zhang and H. Soh, "Large language models as zero-shot human models for human-robot interaction," *arXiv preprint arXiv:2303.03548*, Mar. 2023.

[241] A. Elhafsi, R. Sinha, C. Agia, E. Schmerling, I. A. Nesnas, and M. Pavone, "Semantic anomaly detection with large language models," *Autonomous Robots*, pp. 1–21, Oct. 2023.

[242] Z. Gu, B. Zhu, G. Zhu, Y. Chen, M. Tang, and J. Wang, "Anomalygpt: Detecting industrial anomalies using large vision-language models," *arXiv preprint arXiv:2308.15366*, Sep. 2023.

[243] Y. Li, H. Wang, S. Yuan, M. Liu, D. Zhao, Y. Guo, C. Xu, G. Shi, and W. Zuo, "Myriad: Large multimodal model by applying vision experts for industrial anomaly detection," *arXiv preprint arXiv:2310.19070*, Nov. 2023.

[244] T. Ali and P. Kostakos, "Huntgpt: Integrating machine learning-based anomaly detection and explainable ai with large language models (llms)," *arXiv preprint arXiv:2309.16021*, Sep. 2023.

[245] J. Qi, S. Huang, Z. Luan, C. Fung, H. Yang, and D. Qian, "Loggpt: Exploring chatgpt for log-based anomaly detection," *arXiv preprint arXiv:2309.01189*, Sep. 2023.

[246] C. Egersdoerfer, D. Zhang, and D. Dai, "Early exploration of using chatgpt for log-based anomaly detection on parallel file systems logs," 2023.

[247] C. Xu and J. McAuley, "A survey on model compression and acceleration for pretrained language models," in *Proc. AAAI*, vol. 37, no. 9, Sep. 2023, pp. 10 566–10 575.

[248] W. Yang, C. Yang, S. Huang, L. Wang, and M. Yang, "Few-shot unsupervised domain adaptation via meta learning," in *Proc. IEEE ICME*, Jul. 2022, pp. 1–6.

[249] Y. Liu, Y. Zhu, and J. James, "Resource-constrained federated edge learning with heterogeneous data: Formulation and analysis," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3166–3178, Nov. 2021.

[250] O. Owolabi, "Transfer learning for power outage detection task with limited training data," *arXiv preprint arXiv:2305.17817*, Sep. 2023.

[251] M. L. Di Silvestre, P. Gallo, J. M. Guerrero, R. Musca, E. R. Sanseverino, G. Sciumè, J. C. Vásquez, and G. Zizzo, "Blockchain for power systems: Current trends and future applications," *Renewable and Sustainable Energy Reviews*, vol. 119, p. 109585, Mar. 2020.