Toward Edge General Intelligence With Multiple-Large Language Model (Multi-LLM): Architecture, Trust, and Orchestration

Haoxiang Luo, *Graduate Student Member, IEEE*, Yinqiu Liu, *Member, IEEE*, Ruichen Zhang, *Member, IEEE*, Jiacheng Wang, *Member, IEEE*, Gang Sun, *Senior Member, IEEE*, Dusit Niyato, *Fellow, IEEE*, Hongfang Yu, *Senior Member, IEEE*, Zehui Xiong, *Senior Member, IEEE*, Xianbin Wang, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*— Edge computing enables real-time data processing closer to its source, thus improving the latency and performance of edge-enabled AI applications. However, predictive AI models often fall short when dealing with complex, dynamic tasks that require advanced reasoning and multimodal data processing. This survey explores the integration of multi-LLMs (Large Language Models) to address these challenges in edge computing, where multiple specialized LLMs collaborate to enhance task performance and adaptability in resource-constrained environments. We review the transition from conventional edge AI models to single LLM deployment and, ultimately, to multi-LLM systems. The survey discusses enabling technologies such as dynamic orchestration, resource scheduling, and cross-domain knowledge transfer that are key for multi-LLM implementation. A central focus is on trusted multi-LLM systems, ensuring robust decision-making in environments where reliability and privacy are crucial. We also present multimodal multi-LLM architectures, where multiple LLMs specialize in handling different data modalities, such as text, images, and audio, by integrating their outputs for comprehensive analysis. Finally, we highlight future directions, including improving resource efficiency, trustworthy governance multi-LLM systems, while addressing privacy, trust, and robustness concerns. This survey provides a valuable reference for researchers and practitioners aiming to leverage multi-LLM systems in edge computing applications.**

***Index Terms*— Large language model (LLM), multiple LLMs, edge computing, trustworthy LLM system, multimodal LLM.**

Received 1 July 2025; revised 30 August 2025; accepted 16 September 2025. Date of publication 22 September 2025; date of current version 10 December 2025. This research is supported by Seatrium New Energy Laboratory, Singapore Ministry of Education (MOE) Tier 1 (RG87/22 and RG24/24), the NTU Centre for Computational Technologies in Finance (NTU-CCTF), and the RIE2025 Industry Alignment Fund - Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR. The associate editor coordinating the review of this article and approving it for publication was W. Zhang. *(Corresponding author: Gang Sun.)*

Haoxiang Luo, Gang Sun, and Hongfang Yu are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: lhx991115@163.com; gangsun@uestc.edu.cn; yuhf@uestc.edu.cn).

Yinqiu Liu, Ruichen Zhang, Jiacheng Wang, and Dusit Niyato are with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: yinqiu001@e.ntu.edu.sg; ruichen.zhang@ntu.edu.sg; jiacheng.wang@ntu.edu.sg; dniyato@ntu.edu.sg).

Zehui Xiong is with the School of Electronics, Electrical Engineering and Computer Science (EEECS), Queen's University Belfast, BT7 1NN Belfast, U.K. (e-mail: z.xiong@qub.ac.uk).

Xianbin Wang is with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 5B9, Canada (e-mail: xianbin.wang@uwo.ca).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/TCCN.2025.3612760

I. Introduction

A. Background

EDGE computing has emerged as a crucial paradigm to bring intelligent services closer to data sources (sensors, cameras, vehicles, drones) in networks [1]. By processing data at or near the network edge, computational data exchange-related latency and bandwidth usage can be greatly reduced, thus improving quality-of-service for many time-sensitive intelligent applications such as traffic management, emergency response, and autonomous navigation [2]. Traditional edge Artificial Intelligence (AI) systems, however, typically rely on specialized narrow models, each designed for a specific task, e.g., object detection or traffic prediction [3]. While effective for targeted problems, these conventional models lack the flexibility and general reasoning capabilities required to address the increasingly complex and dynamic environments, such as in modern urban and aerial ecosystems.

In contrast, large pre-trained models, especially large language models (LLMs), have demonstrated remarkable human-level proficiency in understanding, generating, and reasoning over natural language and multimodal data [4], [5], [6]. Their generalist intelligence enables them to perform a wide variety of tasks without task-specific retraining, making them highly attractive for diverse edge applications where versatility and adaptability are essential [7], [8], [9]. As a result, the integration of LLMs with edge computing leads to the emerging concept of Edge General Intelligence (EGI) [10], where edge nodes, such as Internet of Things (IoT) devices, vehicles, drones, and city infrastructure, gain enhanced context awareness, reasoning capabilities, and multimodal interaction [2], [11], [12]. This evolution enables smarter, more autonomous, and adaptive services directly at the edge, reducing reliance on cloud connectivity and accelerating decision-making processes. However, the deployment of LLMs at the edge introduces unique challenges, including

limited on-device compute resources, energy constraints, and high communication overhead for cloud interactions [13], [14]. To alleviate these difficult problems, a few solutions have been recently developed. Researchers adopted techniques such as model compression [15], pruning, quantization, and knowledge distillation [16] to reduce the model size and reasoning complexity to adapt to edge devices. Furthermore, some works have designed energy-saving reasoning, scheduling, and dynamic model invocation strategies [17], [18] rationally allocating tasks to extend the battery life of the equipment. Furthermore, methods such as edge-local inference enhancement and edge-cloud collaborative computing have been proposed to deal with high communication overhead [19].

Just as LLMs empower edge computing, moving towards EGI, researchers have found that there are still limitations when individual LLMs work. Different LLMs, perhaps with different training data, languages, or specialties, can complement each other [20]. Indeed, answers from different LLMs to the same query often vary due to their diverse training corpora and model biases [21]. A single LLM faces difficulty in adapting to heterogeneous contexts (e.g., a traffic management LLM might not handle medical queries) [22]. Additionally, individual LLMs can suffer from outdated knowledge or hallucinations if their training data is limited [23], [24]. Multi-LLM systems have been proposed as a way to advance the state-of-the-art in EGI by having multiple LLMs collaborate. We can combine their strengths and mitigate individual weaknesses by harnessing multiple LLMs as an ensemble or a network [25], [26]. For example, Wang et al. leveraged a pool of LLMs (including GPT-3.5, GPT-4, LLaMA variants, and others) to jointly generate comprehensive elderly care plans, outperforming any single model in coverage of topics [27]. Likewise, researchers have begun exploring multi-LLM systems to simulate "group intelligence," where LLM-based agents discuss or vote on answers to improve reliability [28]. It is also regarded as the prerequisite foundation for a multi-agent system [29]. Although these efforts are still in their infancy, they indicate a paradigm shift toward collaborative intelligence at the edge.

Facing the extensive demand for EGI and recognizing the unique advantages provided by multi-LLM, this paper provides a comprehensive survey of multi-LLM in implementing ubiquitous EGI applications. Before starting the discussion, we summarize the relevant key features of EGI, multi-LLM, and multi-LLM through Fig. 1. The following content will elaborate on this in sequence.

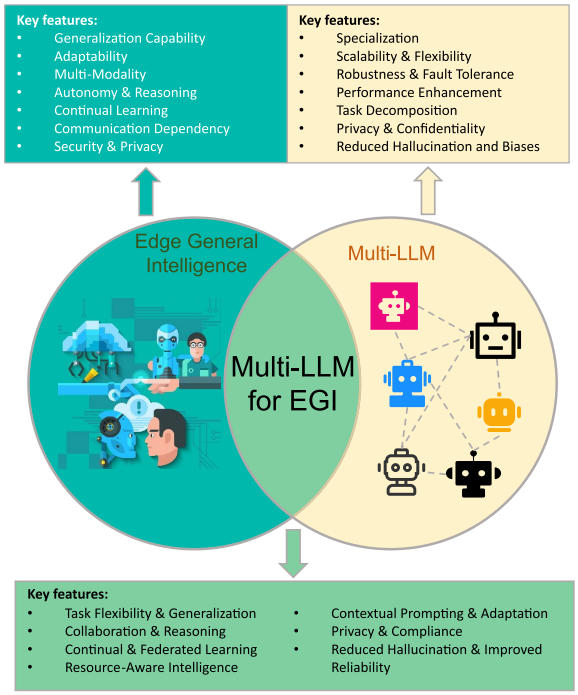

Fig. 1. Key features for EGI, multi-LLM, and multi-LLM for EGI.

B. Related Surveys

To clarify the coverage of the existing relevant surveys and highlight our uniqueness, we have summarized the contributions and priorities of the related work. They are divided into two aspects: LLM for EGI and multi-LLM, which are concluded in TABLE I.

1) LLM for EGI: Recent literature has increasingly investigated the deployment of LLMs at the network edge to enhance intelligent services with reduced latency and bandwidth consumption. The work in [2] presented a comprehensive framework for efficient LLM inference on edge devices by leveraging model compression and hardware acceleration, enabling real-time natural language processing under constrained resources. Dong et al. [30] focused on adaptive model pruning and quantization techniques to reduce computational overhead while maintaining inference accuracy, which is crucial for resource-limited edge environments. Furthermore, Bhardwaj et al. [31] discussed the computing requirements, energy efficiency, and model scalability of deploying LLMs on edge devices with limited resources. Meanwhile, they emphasized the transformative potential and future impact of the combination of LLM and edge computing. In addition, authors proposed a comprehensive overview of the latest advancements in edge LLMs [32]. This work covers the entire life cycle of edge LLMs, from the model design and pre-deployment strategies to runtime inference optimization. The study in [33] explored small language models (SLM), model compression techniques, inference optimization strategies, and dedicated frameworks for edge deployment of LLM. Security and privacy aspects are investigated in [34], where encrypted model updates and differential privacy mechanisms are introduced for secure LLM deployment on heterogeneous edge devices. Meanwhile, Hadish et al. [35] focused on dynamic resource allocation algorithms to adaptively schedule LLM inference tasks across edge clusters, enhancing throughput and energy efficiency. Lastly, Lin et al. [36] systematically surveyed the challenges of LLM deployment at scale on the edge, including hardware heterogeneity, model scalability, and context-aware adaptation, setting a roadmap for future edge intelligence research.

2) Multi-LLM Systems: In [25], multiple LLMs interact and cooperate to overcome individual limitations, enabling enhanced reasoning, knowledge sharing, and multi-task capabilities. It also discusses the key challenges and future

TABLE I
SUMMARY OF RELATED SURVEYS

Scope	Ref.	Overview	LLM	Multi-LLM	Edge computing
LLM for EGI	[2]	A survey on three conceptual architectures of LLM-licensed EGI: centralized, hybrid, and decentralized, and their implementation methods	✓	✗	✓
	[30]	An overview of an efficient memory fine-tuning and model compression in LLM to facilitate its deployment at the network edge	✓	✗	✓
	[31]	A survey emphasizing the role of LLM in reducing latency, enhancing privacy, and improving efficiency for edge computing	✓	✗	✓
	[32]	A comprehensive review of edge LLM from resource-efficient model design, pre-deployment strategies, to runtime inference optimization	✓	✗	✓
	[33]	A comprehensive survey on edge LLM deployment technologies, including small language models, model compression, inference optimization, and frameworks	✓	✗	✓
	[34]	A contemporary survey on the structure, caching, delivery, training, and inference of deploying LLMs by mobile edge intelligence	✓	✗	✓
	[35]	A survey exploring the research trends, developments, and applications of compact edge LLMs	✓	✗	✓
	[36]	A position paper on the deployment of multi-modal LLMs at the 6G edge, including technologies and applications	✓	✗	✓
	[37]	A survey on exploring to reveal and understand how the world model enables EGI	✓	✗	✓
Multi-LLM	[25]	A comprehensive survey on different multi-LLM collaboration strategies such as merge, ensemble, and cooperate	✓	✓	✗
	[38]	A survey on the integration of multiple LLMs before, during, and after reasoning	✓	✓	✗
	[39]	A position paper emphasizing the multi-LLM collaboration to address the challenges in single LLM, such as reliability, democratization, and diversity	✓	✓	✗
	[40]	A survey on the relationship (cooperation and competition) between large and small language models	✓	✓	✗
	[41]	A conceptual framework on using blockchain to enable multiple LLMs to collaborate and serve wireless networks	✓	✓	✗
	[42]	A review on two complementary reasoning strategies between multi-LLM systems, including routing, and cascading	✓	✓	✗

opportunities in coordinating such multi-model systems. Then, Chen et al. [38] comprehensively reviewed the emerging field of LLM ensembles, categorizing methods into ensemble-before-inference, ensemble-during-inference, and ensemble-after-inference paradigms. It analyzed various techniques, related challenges, benchmarks, and applications, highlighting how combining multiple LLMs can improve performance beyond single-model usage. Meanwhile, in the position from [39], it argued that a single LLM is insufficient to reliably represent the diverse and complex real-world data, skills, and user populations, advocating instead for multi-LLM collaboration. Moreover, Chen and Varoquaux [40] analyzed the complementary roles of small models alongside LLMs. It highlighted how SLMs contribute to data curation, efficient inference, interpretability, and domain-specific tasks, thus promoting resource-efficient and practical AI deployments. Finally, Luo et al. [41] proposed a blockchain-enabled trustworthy multi-LLM network framework to enable collaborative, secure, and reliable responses from multiple LLMs for complex wireless network optimization. It also demonstrated an effective defense mechanism against false base station attacks in 5G or 6G wireless systems.

Recent investigations have studied the integration of LLMs in edge computing or the work on the collaboration of multiple LLMs. However, up to now, there has been no comprehensive investigation into the unique opportunities and challenges of multiple LLMs working together in edge networks to achieve EGI. In this survey, our goal is to fill this gap. We have delved into the review of the architecture and technology of multi-LLM. These techniques can achieve powerful and reliable edge intelligence in ubiquitous scenarios.

C. Our Contributions

The key contributions of this paper are summarized as follows:

- We provide a comprehensive review of architectural designs and deployment strategies tailored for multi-LLM systems operating in edge computing environment. We also investigate the differences among predictive AI, single LLM, and multi-LLM systems. We comprehensively discuss the differences between running multi-LLM in the cloud and on the edge. Furthermore, we summarize four typical edge applications to reveal the potential of multi-LLM, especially in mobile scenarios such as elderly care, smart grid inspection, intelligent transportation, and Low-Altitude Economic Networks (LAENets).
- This survey categorizes key enabling technologies based on model compression, resource orchestration, model context protocol, privacy protection, LLM fine-tuning, and multimodal information fusion. This work lays a foundation for designing a robust multi-LLM-enabled

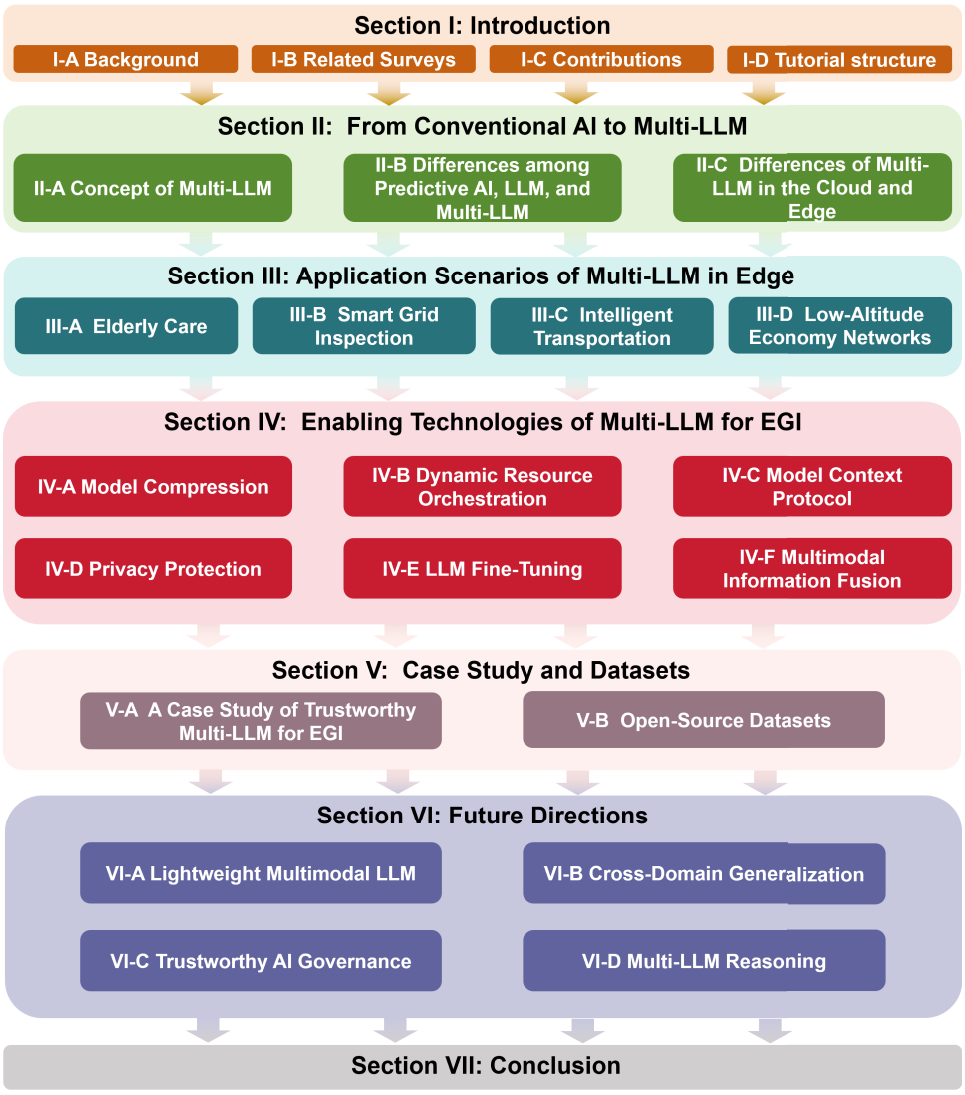

Fig. 2. Structure of our survey.

ubiquitous EGI that maximizes local intelligence while reducing reliance on centralized cloud processing.

- We provide a case study and an open-source dataset summary for deploying multi-LLM systems at the edge. Trustworthiness is a critical factor, where decisions often impact safety-critical or privacy-sensitive applications. We conduct an in-depth analysis of the mechanisms that ensure reliability and security within multi-LLM collaborations. Then, the available datasets widely used in multi-LLM systems are conducive to promoting further research and experiments in EGI.
- Future multi-LLM systems should design lightweight multimodal models that efficiently handle diverse data under edge resource constraints. Therefore, enhancing cross-domain generalization to support knowledge transfer and dynamic adaptation is also essential. Next, establishing a trustworthy AI governance ensures transparency, fairness, securing critical applications. Moreover, the reasoning methods in the multi-LLM system are crucial for further enhancing its ability to handle tasks. Addressing these challenges will enable a strong multi-LLM system, driving advancements in ubiquitous EGI.

D. Structure of This Paper

The structure of this survey is outlined in Fig. 2. Section II introduces the concept and design motivation of multi-LLM, including its differences from predictive AI and individual LLM. In Section III, it demonstrates four typical applications of multi-LLM. Then, Section IV illustrates how edge computing moves towards EGI, that is, the key enabling technologies of multi-LLM empowering EGI. Next, as a tutorial, Section V provides a case study on the trustworthy multi-LLM for EGI and summarizes open-source datasets that support multi-LLM research. Moreover, Section VI reveals the future directions of multi-LLM for edge computing. Finally, Section VII summarizes this survey.

II. From Conventional AI to Multi-LLM

This section introduces the multi-LLM concept, including the collaboration methods among LLMs, as well as

the differences among predictive AI, individual LLMS, and multi-LLM. The application scenarios of multi-LLM are also discussed.

A. Concept of Multi-LLM

Multi-LLM systems refer to architectures or frameworks where multiple LLMs operate together (in parallel or in sequence) to solve tasks or generate outputs. In essence, rather than relying on a single monolithic model, a multi-LLM system leverages an ensemble or team of LLMs that work in concert by sharing work or validating each other's outputs [25]. This collaboration allows each model to contribute its specialized strengths toward a shared objective, such as domain knowledge, language, or reasoning skills.

A key aspect of multi-LLM systems is how the different models collaborate or interact. Several collaboration modes have been explored in technical literature, each with distinct philosophies for organizing multiple LLMs. The major paradigms include cooperative approaches, competitive (or adversarial) approaches, and various ensemble-based schemes. In practice, a multi-LLM system might combine elements of these modes to achieve a balance. Table II summarizes the representative works of these different types of LLM collaborations. Below, we outline each collaboration mode, clarifying what it entails and providing examples from recent frameworks.

1) Cooperative Collaboration: Cooperative collaboration describes scenarios where multiple LLMs work synergistically with aligned goals, often dividing complex tasks into subtasks or complementing each other's strengths. This paradigm leverages role specialization, information sharing, and iterative refinement to enhance overall performance and robustness.

For instance, Owens et al. [21] investigated centralized and decentralized dialogue frameworks enabling multi-LLM communication to collectively reduce bias and enhance fairness. Dai et al. [43] developed an online multi-LLM selection strategy that dynamically allocates queries to different LLMs based on cost-performance trade-offs, facilitating efficient resource utilization in real time.

In robotics and multi-agent systems, Mahadevan and Peternel [44] demonstrated cooperative LLM-based navigation agents that communicate via natural language to resolve conflicts and assign priorities. Yuan et al. [45] integrated human reviewers into a multi-LLM consensus pipeline to improve annotation quality while reducing manual effort. Mahadevan et al. [46] proposed hierarchical multi-LLM conversational agents that cooperate across different processing levels to improve dialogue efficiency and user experience. Notably, Feng et al. [47] introduced the modular pluralism framework, enabling token-level collaboration between a large closed-box LLM and multiple community-tuned LLMs, facilitating pluralistic alignment that better represents diverse societal values.

To summarize, cooperative systems are particularly effective in contexts requiring fairness, bias mitigation, contextual reasoning, and multimodal fusion, providing complementary knowledge and cross-verification capabilities.

2) Competitive Adversary: Not all multi-LLM interactions are purely cooperative. Some frameworks introduce competition or adversarial dynamics between models. In a competitive mode, each LLM prioritizes its objective or hypothesis, which may conflict with others' objectives [54]. Rather than helping each other outright, the models essentially strive to outperform or correct one another, with the system ultimately picking the best outcome from the contest. The rationale is that competition can push each model to perform at its best, yielding a more rigorous solution overall.

A case of competition is the adversarial collaboration approach, where models are explicitly set to challenge each other's outputs in order to stress-test the correctness and consistency of results [55]. In adversarial debate frameworks, multiple LLMs serve as "advocates" or debaters for different answers, and they iteratively critique each other's reasoning [56]. This approach essentially transforms the problem-solving process into a dynamic dialogue among LLMs that validates and cross-examines potential answers.

For example, Feng et al. [24] utilized this approach by enabling multiple LLMs to adversarially critique uncertain responses produced by their peers, mitigating hallucinations and improving answer fidelity. Socratic multi-LLM dialogue [48] similarly incorporates adversarial debates, allowing models to refine outputs through dialectical conflict, resulting in enhanced reasoning depth and robustness.

In summary, competitive or adversarial multi-LLM modes introduce a "checks and balances" effect. Each model's output is vetted by others, promoting robustness at the cost of a more complex interaction protocol.

3) Ensemble Integration: Ensemble-based methods are a classical approach adapted to LLMs. Multiple models are run in parallel on the same task, and their outputs are combined to produce a final result. Unlike the interactive cooperation or debate methods above, a pure ensemble often involves little to no direct communication between the LLMs during inference. Each model independently generates an answer (or prediction), and then a predefined aggregation mechanism synthesizes these results [25]. The goal is to harness the individual strengths of each model and cancel out their independent errors, improving overall accuracy or reliability.

Typical integration relies on the voting results of each LLM. On the one hand, the average probability distribution of different results can be statistically analyzed from multiple LLMs to determine the final response. For example, Luo et al. [41] used voting to decide on the best base station power allocation scheme to resist fake base station attacks for the wireless communication system. On the other hand, there is the weighted voting of confidence level. If the models can provide confidence scores for their answers, the final decision will tend towards the answer with the highest total weight. Luo et al. [41] designed a weighted Byzantine fault tolerance to ensure robustness of multi-LLM in wireless network optimization.

In addition, some other typical examples of integration, including that Fang et al. [50] examined ensemble strategies for text summarization, contrasting centralized and decentralized frameworks. Mao et al. [51] presented a statistical deep ensemble framework tailored for detecting AI-generated

TABLE II
COMPARISON OF MULTI-LLM WORKS BY COLLABORATION TYPE

Ref.	Contributions	Key Techniques	Type
[21]	Centralized and decentralized multi-LLM dialogue frameworks reduce bias and improve fairness and accuracy	Multi-model interaction, bias detection, decentralized communication	■
[43]	Online multi-LLM selection algorithm dynamically balances performance and cost for task scheduling	Multi-armed bandit algorithm, task scheduling, online learning	■
[44]	Multi-robot navigation via natural language communication, resolving conflicts and prioritizing tasks	Multi-agent dialogue, game theory, control barrier functions	■
[45]	Multi-LLM consensus with human review framework improves annotation accuracy and reduces human effort	Independent multi-model analysis, consensus mechanism, human-AI collaboration	■
[46]	Multi-level hierarchical LLM conversational agent improves robotic interaction and user satisfaction	Multi-level query classification, hierarchical dialogue management, speech recognition	■
[47]	Modular pluralism framework uses multi-LLM collaboration for pluralistic alignment	Fine-tuning, token-level multi-LLM interaction, modular decoding strategies	■
[24]	Multi-LLM competition improves QA accuracy and confidence, effectively reducing hallucinations	Multi-model reflective reasoning, confidence estimation	♦
[48]	Socratic multi-LLM dialogue framework enhances reasoning quality through debate and opposing viewpoints	Multi-turn dialogue, debate strategies, reasoning evaluation	♦
[49]	A multiple LLM debate theoretical framework, along with three improvement methods: diversity pruning, quality pruning, and rebuttal of misunderstandings	Context learning, Bayesian reasoning, debate strategies, pruning	♦
[28]	Blockchain-enabled trusted multi-LLM with weighted Byzantine fault tolerance consensus enhances trust and robustness	Blockchain consensus, decentralized trust evaluation	●
[41]	Blockchain-enabled trustworthy multi-LLM network mitigates biases and malicious behavior in wireless network optimization	Blockchain consensus, P2P (Peer to Peer) network, multi-LLM cooperative reasoning	●
[50]	Multi-LLM text summarization framework compares centralized and decentralized strategies, enhancing quality	Centralized evaluation, distributed evaluation, summary generation	●
[51]	Multi-LLM statistical deep ensemble framework for high-precision AI-generated Chinese text detection	Mixture of Experts (MoE), statistical feature extraction, cross-entropy metrics	●
[52]	Sampling-simulation method improves offline multi-LLM inference efficiency with dynamic scheduling and parallelism	Sampling simulation, greedy scheduling, multi-GPU parallelism	●
[53]	A Multi-LLM Knowledge Fusion (MLKF) for simulating the cognitive and reasoning process of humans	Cognitive reasoning, knowledge fusion	●

■: cooperative collaboration; ♦: competitive adversary; ●: ensemble integration

Chinese text with high precision, employing Mixture of Experts (MoE) and statistical feature extraction. Meanwhile, to accelerate the integration of multi-LLM, Fang et al. [52] introduced a sampling-simulation method to improve multi-LLM offline inference efficiency, leveraging dynamic scheduling and parallelism across multiple GPUs.

In general, ensemble methods benefit from the diversity of individual models. Also, it typically yields stable and reliable outputs without tightening model interaction during inference.

B. Differences Among Predictive AI, Single LLM, and Multi-LLM

Before applying the multi-LLM system to edge computing, it is necessary to understand its characteristics and advantages. In this part, we compare the differences among predictive AI models, single LLM, and multi-LLM. Fig. 3 reveals in detail their characteristics, typical cases, etc.

1) Predictive AI: Predictive AI models encompass classical machine learning algorithms and compact neural networks that are tailored to specific tasks. These models are typically purpose-built. Each model is trained on a well-defined dataset to perform a particular function, e.g., object detection, anomaly detection, or signal classification. Common designs include decision trees, support vector machines [57], and small-scale deep neural networks such as lightweight Convolutional Neural Networks (CNNs) [58] or shallow Multi-Layer Perceptrons (MLPs) [59]. They are selected because they operated efficiently on the limited-resource devices. The underlying design philosophy is to keep models narrow and optimized for the target task, using hand-crafted features or specialized network architectures to maximize accuracy within that domain [60], [61]. In terms of capabilities, traditional models excel at the tasks they are designed for, often achieving high performance under the conditions they are trained on.

The reasoning and generalization capabilities of these AI models are limited. Because they cannot infer beyond their training distribution, nor do they have the ability to handle tasks beyond their narrow authorization [62]. For example, a CNN-based traffic classifier can categorize known traffic signs accurately. But it cannot interpret natural language instructions or adapt to recognize new, unrelated objects without retraining. Each additional functionality (say, adding voice command recognition to a camera system) typically requires developing or deploying a separate model or algorithm.

Due to their limited scope, predictive AI is usually organized in isolated or pipeline structures rather than as an integrated system [63]. Especially at the edge, deployment may involve multiple independent models. Each model handles one aspect of complex tasks. For example, one model is used for face recognition, and another model is used for audio detection. However, any advanced decision logic must be manually encoded or processed by a simple rule-based framework. Therefore, traditional edge artificial intelligence

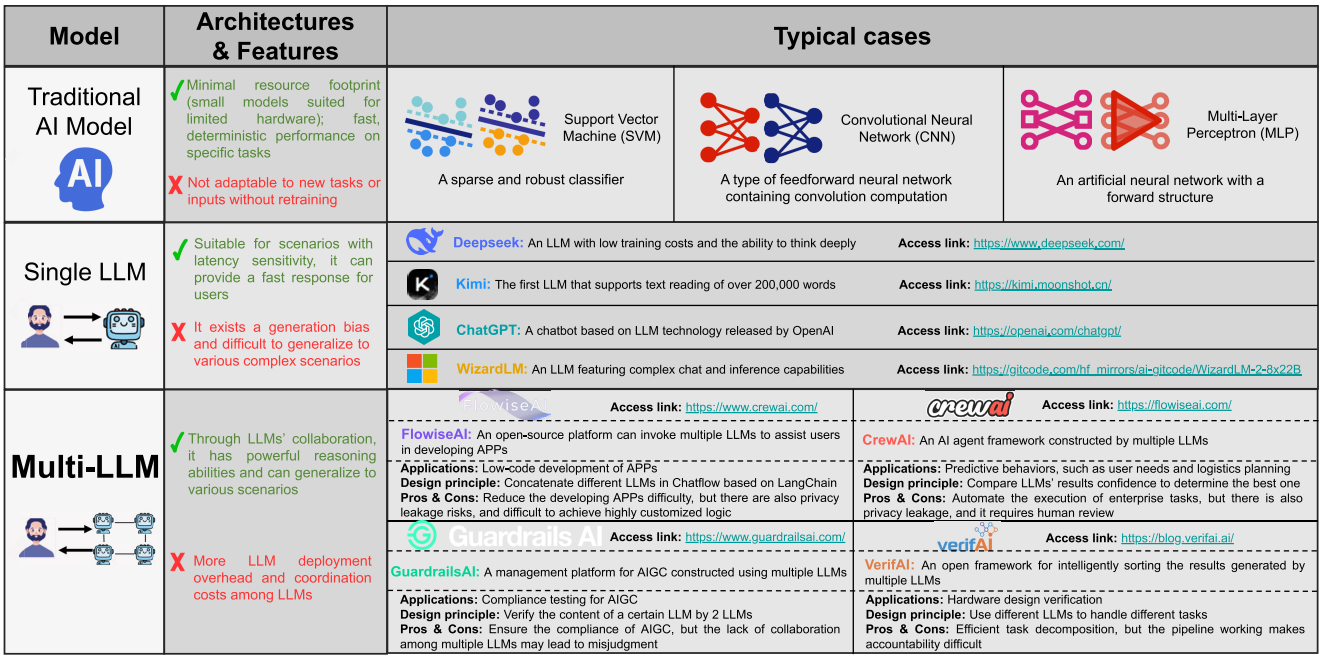

Fig. 3. Comparison with predictive AI, single LLM, and multi-LLM system. Predictive AI models require targeted training based on the particularity of scenarios and lack generalization. Due to the limitations of training techniques and data, a single LLM often generates biased and illusory results. Multi-LLM can overcome the above problems through the collaboration of multiple LLMs.

cannot learn dynamically from each other's outputs in an open manner [64].

2) Single LLM: Single LLM refers to a general-purpose intelligence may based on the Transformer architecture [65]. Typical representatives include ChatGPT,[1] WizardLM,[2] Deepseek,[3] and Kimi,[4] etc. These models are characterized by deep layers of self-attention and very large parameter counts, having been pre-trained on massive text or multi-modal corpora [66]. Notably, these models learn a broad statistical representation of language and world knowledge during training. Their underlying design leverages unsupervised or self-supervised learning at scale, which endows them with processing capabilities far beyond any single narrow task model [67].

A single LLM at the edge can serve as a versatile AI agent handling numerous tasks through prompting or few-shot examples, rather than through task-specific reprogramming [68]. Moreover, LLMs demonstrate a degree of general reasoning [69]. They can solve problems they have not been explicitly trained on by drawing on analogous examples from their training knowledge. For example, an edge-deployed LLM can analyze log data or sensor readings expressed in natural language and provide insights or summaries [70], then switch to answering users' questions. It is something traditional models cannot do, as LLMs effectively carry a broad prior learned from diverse data. Such cognitive flexibility is highly attractive for edge intelligence scenarios where the types of queries and data can vary widely [71].

[1]https://chatgpt.com/
[2]https://github.com/nlpxucan/WizardLM
[3]https://www.deepseek.com/
[4]https://www.kimi.com/

Deploying a single LLM on edge hardware, however, is challenging due to resource requirements [72]. These models typically demand gigabytes of memory and significant compute, for example, tens of billions of parameters require specialized hardware accelerators [73]. In practice, making an LLM edge-friendly involves model compression techniques like quantization and distillation [74], reducing precision or size to fit on GPUs or CPUs available at the edge. Recent works introduce SLMs [33], which are trimmed-down LLMs, on the order of 0.5-7 billion parameters, that aim to retain much of the original's capability while running on a single edge device. Even so, the inference cost is substantial. The optimized models often still need at least on the order of 500 MB of RAM and an advanced processor to run effectively [73]. This can strain devices such as IoT gateways or smartphones, especially under real-time constraints. Edge deployments of LLMs may thus require hardware upgrades, e.g., adding an AI accelerator module or offloading parts of the computation to nearby edge servers [75]. Techniques like split inference, that is, partitioning the model layers between device and cloud, are also explored to cope with this demand, albeit introducing complexity in return [76].

Despite the remarkable capabilities of single LLMs, their use in edge computing exposes critical limitations. Individual models often produce hallucinations, generating biased or false content, due to incomplete training data or architectural constraints [21], [77], [78]. These issues not only hinder reliability and adaptability in dynamic network settings but also highlight the single-model approach as a potential bottleneck [32]. Consequently, researchers are turning to multi-LLM collaborative systems to overcome these shortcomings. This paradigm shift, as detailed next, forms a logical step

toward more resilient and scalable intelligence at the network edge.

3) Multi-LLM System: Multi-LLM collaboration involves a network of LLM-based agents working together, potentially offering a form of collective intelligence. Through various types of collaboration methods, namely collaboration, competition, and ensemble, they have avoided the generalization weaknesses, such as illusion and bias of individual LLMs [21], [24]. This approach finds typical use cases in 360 AI Assistant[5] and Corex.[6] The former is capable of triggering the collaborative operation of three LLMs. Developed by the Shanghai AI Lab, the latter allows multiple LLMs to carry out reasoning in a joint manner. Moreover, Amazon[7] has delved into the field of message routing approaches among multiple LLMs. Additional instances are presented in Fig. 3.

The key advantages of the multi-LLM system framework can be summarized as follows:

- **Specialization:** The capabilities of a multi-LLM system arise from the diversity and specialization of its member models. By leveraging different knowledge bases and strengths of each LLM, such a system can cover a broader range of tasks or a more complex task space than any single model. For example, one LLM could be a math expert while another is skilled in understanding legal text. Through collaboration, they can solve problems that involve both legal interpretation and numerical reasoning [79].
- **Scalability & Flexibility:** Compared with a single LLM, expansion usually means retraining or fine-tuning the entire model. For multi-LLM systems, new models can be added or replaced without retraining the entire system, thereby adapting to new tasks.
- **Robustness & Fault Tolerance:** In essence, multi-LLM systems introduce redundancy and specialization. They are more fault-tolerant and can tackle multifaceted tasks by distributing subtasks among themselves. This approach is inspired by multi-agent system principles, moving AI from isolated models to a collaboration-centric paradigm [54].
- **Performance Enhancement:** Even when models are of a similar general-purpose nature, collaboration allows for cross-verification and ensemble effects. An LLM can double-check or critique the output of another, similar to having multiple advisors discuss a problem, which often yields a more accurate and robust result [48].
- **Task Decomposition:** The system's architecture can take various forms. A centralized scheme might use a leader or supervisor LLM that delegates subtasks to other specialist LLMs. On the contrary, to avoid a single point of failure, a decentralized scheme treats each model as a peer in a distributed protocol, for example, voting or consensus to ensemble results.
- **Privacy & Confidentiality:** Through the hierarchical data management of multi-LLM, users' sensitive data can be routed to be stored in a private or secure LLM. While public or non-privacy-sensitive types of queries can use the public model. In a single LLM, all data has to pass through the same LLM, which raises concerns about privacy and compliance. Moreover, LLM can manipulate data without the knowledge of other entities [80].
- **Reduced Hallucination and Biases:** Due to the limitations of a single LLM in terms of training data, technical paths, etc., the generated content may be outdated, biased, and illusory. A system composed of multiple LLMs can effectively avoid this problem, provide users with comprehensive responses, and generalize to various complex scenarios [28], [41].

It is precisely because of these powerful capabilities, the enhanced reasoning ability, and the more comprehensive information interaction through collaboration among LLMs, that multi-LLM systems are also seen as evolving towards agentic AI [81]. As a result, multi-LLM can bring benefits to EGI, such as task flexibility and generalization, collaborative reasoning, and reduced hallucinations. However, it also introduces many complex challenges. First, the resource footprint grows with each additional model. Running several LLMs in parallel can easily exceed the capacity of the edge side. One solution is to distribute the models across multiple devices, forming an edge cluster [82], but this introduces network communication overhead and requires synchronization. Second, coordination mechanisms must be implemented. Unlike a single model, multi-LLM systems need an agreed-upon protocol for collaboration [83]. This might be an internal messaging schema, a turn-taking conversation, or an algorithm for consensus on final answers. Third, there are also trust and consistency challenges. In a decentralized edge setting, how do we trust that each model is providing honest, correct information? Malfunctioning or malicious LLMs could poison the collaborative process. As a result, developing and maintaining multi-LLM systems is still an open research area.

C. Differences of Multi-LLM in the Cloud and Edge

Mobile edge deployments of multi-LLM systems differ markedly from cloud deployments across several dimensions. Inference latency is typically lower at the edge because data is processed close to its source, avoiding long network backhauls. By contrast, cloud-based LLM inference incurs additional round-trip delays to distant data centers [84].

Resource constraints are far more pronounced in edge environments. Edge servers have limited compute, memory, and power, making it challenging to host LLMs. For instance, a 7 B-parameter LLaMA 2 requires 28 GB RAM, exceeding most edge devices' capacity without compression or partitioning [84]. Cloud data centers can leverage abundant GPUs and memory to run such large models more easily.

Consequently, the system architecture tends to be distributed in edge scenarios. Model inference may be split across multiple edge nodes or performed collaboratively between devices and edge servers [85]. In contrast to the centralized architecture in clouds, the entire model ensemble runs in one location. These trade-offs inform applicability. Real-time

[5]https://bot.360.com

[6]https://link.zhihu.com/?target=https%3A//github.com/QiushiSun/Corex

[7]https://aws.amazon.com/cn/blogs/machine-learning/multi-llm-routing-strategies-for-generative-ai-applications-on-aws/

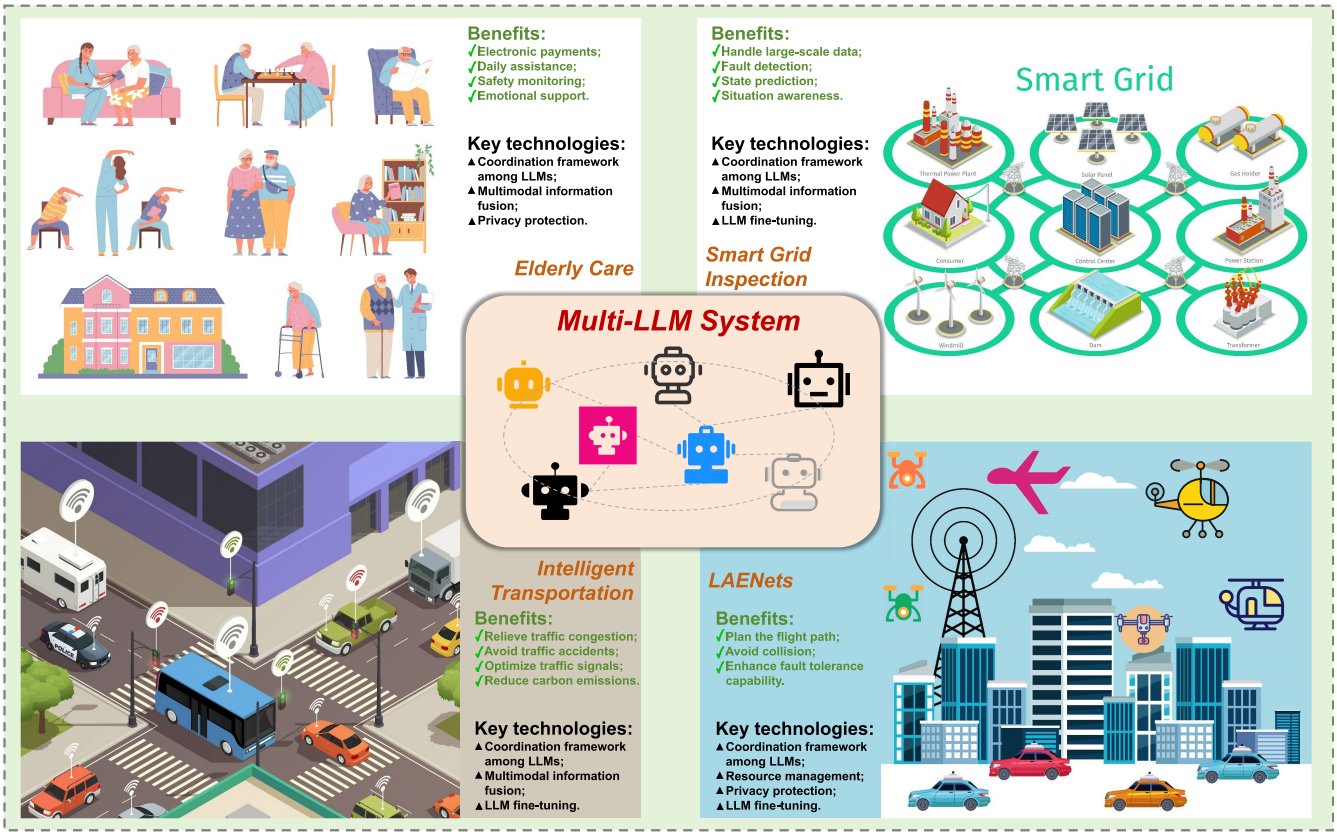

Fig. 4. Application scenarios of multi-LLM. It includes typical edge scenarios such as elderly care, smart grid inspection, intelligent transportation, and LAENets.

and privacy-sensitive use cases often favor edge deployments for their immediate responsiveness and local data handling. No need to expose sensitive data to external servers. While cloud-based LLM systems, although they have superior computing capabilities, may have difficulty meeting strict latency needs.

III. Application Scenarios of Multi-LLM

To demonstrate the outstanding capabilities of multi-LLM in edge computing, we list four typical applications here, including elderly care, smart grid inspection, intelligent transportation, and LAENets, as shown in Fig. 4. Especially the latter three, reveal the great potential of multi-LLM in mobile scenarios. The extensive application scenarios of multi-LLM have promoted the realization of ubiquitous EGI.

A. Elderly Care

The multi-LLM system transforms elderly care by serving as smart assistants that address diverse needs in health monitoring, daily living, and social interaction. This application demands multiple LLMs instead of a single model because the tasks span numerous domains (medical advice, emergency alerts, household management, etc.) and modalities (text, speech, sensor data). A single model would be limited in expertise and may have outdated or biased knowledge. While a LLMs team can cover broader knowledge and cross-verify each other's outputs. For example, one recent system combined GPT-3, GPT-4, and six other LLMs to collectively provide personalized elder care services, handling electronic payments, daily assistance, safety monitoring, and emotional support [27]. Such multi-LLM collaboration brings clear benefits. Each LLM can be fine-tuned for a specific function, e.g., a medical domain LLM for health queries, a home automation LLM for smart appliances, a conversational LLM for companionship. By assigning distinct roles to different LLM-based agents and establishing feedback loops, the system can catch errors or hallucinations and improve response quality [86].

Edge deployment characteristics are crucial in this scenario to ensure responsiveness and trust. Elderly care assistants are often deployed on distributed edge devices in smart homes or wearable monitors, which keeps sensitive personal and medical data local [87]. On-device processing with multiple cooperating LLMs avoids sending raw data to the cloud, protecting privacy while reducing latency for time-critical support [88], such as fall detection or medication reminders. The multiple LLMs communicate over the local network or via an edge gateway, requiring efficient coordination protocols to share insights without overwhelming bandwidth. The key supporting technologies include the agent coordination framework, which allows these LLMS to negotiate tasks and share intermediate results. Furthermore, multimodal information fusion requires the integration of camera feeds, biological signals, and natural language inputs to obtain an overall view of the user's state [89]. Equally important is privacy-preserving

collaboration [90]. Techniques such as on-device inference and access control ensure that even as models collaborate [91], the user's personal data remains protected. In this way, collaborative LLMs can provide consistent, accurate, and user-adaptive assistance to the elderly in a trustworthy manner.

The core feature of this scenario is the cross-domain and multimodal integration of tasks, which needs to simultaneously meet various demands such as health monitoring, daily assistance, safety warning, and emotional companionship. Therefore, the multi-LLM system needs to achieve cooperative collaboration among LLMs to address the challenges in elderly care.

B. Smart Grid Inspection

Smart grid inspection tasks benefit greatly from a multi-LLM approach, including monitoring IoT sensor streams and analyzing drone imagery of power lines [92], [93]. Unlike siloed single-purpose tools, multiple LLMs working in concert can holistically reason over heterogeneous data sources. One LLM alone might be overwhelmed by the volume and variety of grid data or miss cross-modal correlations. By contrast, specialized LLM agents can each focus on a particular modality or sub-problem and then share their findings. For instance, an LLM could correlate a series of voltage sensor anomalies with recent weather conditions, while another examines drone photos of a transformer for physical damage. Together, they provide a more complete situational picture and can suggest targeted maintenance actions [94]. Multi-LLM cooperation yields superior situational awareness for grid operators by cross-verifying insights, which reduces false alarms and improves the accuracy of fault detection and prediction.

Edge deployment is essential for this scenario, given the geographic distribution of the grid and the need for real-time response. Placing LLMs on distributed edge, at substations, control centers, or on inspection drones, minimizes communication delays. It can ensure critical functions can continue locally if cloud connectivity is lost [95]. This arrangement significantly cuts fault response times and reduces bandwidth usage. LLMs communicate over reliable networks or dedicated 5G links, to enhance network security and resilience [96]. Key enabling technologies include multimodal data fusion, which combines sensor measurements, weather information and visual inspections into a unified situational awareness. In addition, LLM specialization empowers knowledge of power systems in fine-tuning models [97], enabling them to understand specific grid terms and failure modes. Also, proxy coordination protocols for distributed decision-making are required [98]. For example, the edge LLM of the substation can request confirmation of the LLM of the drones before triggering the alarm. Furthermore, achieving real-time performance requires model optimization and dynamic resource scheduling. Heavy computing can be offloaded to more powerful nearby servers. Through the above optimization, the multi-LLM method can immediately perform grid anomaly detection and carry out dynamic grid reconfiguration at the edge, thereby pointing to a more resilient grid.

Compared with the previous scenario, the smart grid inspection not only adheres to the multi-modal data fusion principle, but also involves aggregating the results generated by multiple LLMs for the same task. For instance, multiple UAVs collect images, etc. Therefore, this scenario requires a multi-LLM system to achieve the cooperation and integration of multiple LLMs.

C. Intelligent Transportation

Multi-LLM deployments offer a powerful paradigm for intelligent transportation systems by enabling autonomous traffic management and connected vehicles to collaborate in real time. Urban mobility involves many distributed actors, such as vehicles, traffic signals, and control centers. In highly dynamic conditions, fixed rule-based systems find it difficult to efficiently coordinate these participants [99]. Additionally, a single centralized AI has limited adaptability and scope, whereas a team of LLMs can divide the cognitive load and respond more flexibly to evolving traffic situations [100]. For example, each smart intersection could be managed by its own LLM optimizing local signal timing. A higher-level coordinator LLM oversees city-wide patterns and mitigates congestion by rerouting flows. Vehicles' LLM coordinates maneuvers with each other and the infrastructure. This multi-LLM collaboration can reduce congestion and accidents, lower emissions, and provide natural language interfaces for human operators and users. Unlike rigid traditional Intelligent Transportation System (ITS) control, LLM can handle novel scenarios, e.g., unexpected road closures, rather than being limited to pre-programmed responses [101].

Edge computing is pivotal in this scenario to meet the ultra-low latency requirements of transportation [102]. Safety-critical functions, such as collision avoidance or emergency braking coordination, demand millisecond response times. Thus, computation must occur close to data sources, on vehicles and Road Side units (RSUs), or nearby edge servers [103]. This distributed architecture ensures each vehicle or intersection makes instant local decisions while still coordinating with other LLMs for cooperative maneuvers [104]. High-speed V2X (vehicle-to-everything) communication enables these LLM agents to exchange information with minimal latency. The key supporting technologies include hierarchical coordination. Some LLMs assume supervisory roles to coordinate the actions of many agents and prevent conflicts [105]. Moreover, multimodal perception integration requires the combination of visual sensor data, maps, and other inputs with LLM inference to better understand the traffic environment [106]. LLM specialization can fine-tune some models to accomplish expert tasks such as route optimization, event analysis, or traveler communication. Multi-LLM can deliver smart mobility services, automate tasks like traffic analysis and simulation. By collaborating through both data-driven and natural language exchanges, distributed LLMs collectively make transportation networks safer, more efficient, and more user-friendly.

Intelligent transportation is more complex than the previous scenario and involves competitive adversaries among different LLMs. Because in an automated traffic flow, different vehicles may simultaneously use the same road. At this time, there is

a debate over the benefits of both parties, as well as considerations by the on-board LLMs regarding potential safety hazards.

D. Low-Altitude Economy Network

LAENets are emerging as ecosystems of aerial platforms, including delivery drones, urban air mobility vehicles, and UAV-based wireless service nodes [107]. They operate in the lower airspace to provide logistics, communication, and sensing services [108]. Managing such a complex network of autonomous agents requires multi-LLM collaboration to handle the breadth of tasks and the highly dynamic environment. A traditional centralized control or single AI model would be hard-pressed to adapt to constantly changing conditions and uncertainties. In contrast, a multi-LLM-powered distributed team can divide responsibilities and operate concurrently. For example, each drone's LLM plans its route on the fly; a network-management LLM optimizes communication links; and a coordinator LLM oversees mission-level objectives while deconflicting routes [109]. This division of labor allows the system to scale and remain robust. Each LLM makes local decisions using specialized domain knowledge, while sharing essential information such as sensing, navigation, communications, delivery, etc. [107], to collectively adapt as conditions change. Multi-LLM cooperation improves safety, negotiating the right-of-way to prevent mid-air collisions. It also increases efficiency, collectively scheduling routes to save energy and time [110]. Meanwhile, it can boost resilience by avoiding single points of failure. For instance, if one LLM encounters a problem, others can assist or reroute tasks, providing fault tolerance beyond any monolithic controller [111].

Edge computing and communication infrastructure are fundamental to deploying multi-LLM systems in LAENets [112], [113]. Each drone is an edge node with limited compute and battery, so only lightweight LLM models run onboard for immediate decisions. Additionally, each drone must act autonomously if a link drops, so local intelligence is indispensable. Key enablers include coordination protocols for LLM-driven drones to share state and plans, such as navigation, communications management, and sensing. Then, security is also critical. Robust authentication can prevent malicious or faulty drones from disrupting the network [114]. Finally, intelligent resource management allocates communication bandwidth, computation, and energy among UAVs, dynamically scheduling tasks and data flows to maximize network performance [115]. By deploying collaborative LLMs at the edge, LAENets become far more adaptive and reliable in supporting next-generation drone delivery and aerial network services.

LAENets is similar to intelligent transportation scenarios and also involves three types of multi-LLM relationships. Furthermore, it introduces more complex variables and considerations in the spatial dimension.

E. Lessons Learned

The application of multi-LLM in mobile edge scenarios shows great potential. However, based on the analysis of the above four scenarios, there are also many challenges, such as resource constraints and high real-time requirements. With the help of key technologies such as model compression, dynamic resource orchestration, model context collaboration, privacy protection, model fine-tuning, and multimodal information fusion, the efficient deployment of multi-LLM in mobile edge scenarios can be achieved, promoting the ubiquitous EGI.

IV. Enabling Technologies of Multi-LLM for Edge General Intelligence

According to the lightweight characteristics for LLM to empower EGI, we classify the key enabling technologies into six categories. The following will elaborate respectively.

A. Model Compression

Edge deployment of LLMs is challenging due to limited device resources, motivating research on lightweight architectures and model compression. Knowledge distillation is a prevalent approach. For example, DistilBERT [116] and TinyBERT [117] are compact models distilled from BERT (Bidirectional Encoder Representations from Transformers) that preserve most of its accuracy. The former retains 97% of BERT's performance with 40% fewer parameters, and the latter achieves 96.8% of BERT-base accuracy while being 7.5× smaller and 9.4× faster. Another strategy is architectural optimization. MobileBERT [118] introduces bottleneck structures and a custom intermediate teacher model, yielding a 4.3× smaller and 5.5× faster variant of BERT that matches its accuracy on many tasks. Similarly, ALBERT [119] reduces model size by factorizing embeddings and sharing parameters across layers, achieving an order-of-magnitude parameter reduction. Specifically, ALBERT-large has 18 M vs 334 M parameters in BERT-large with minimal loss in accuracy. Other techniques include pruning redundant weights and quantizing model weights to reduce the memory footprint and inference latency. EdgeBERT [120] exemplifies a combination of these. It uses adaptive attention, selective pruning, and quantization to fit BERT on edge devices, yielding up to 7× energy savings in multi-task NLP (Natural Language Processing) inference. Collectively, these innovations drastically reduce LLM model size and computation, enabling LLM deployment within the strict latency and memory constraints of edge computing.

Beyond single-model compression, lightweight multi-LLM architectures at the edge often coordinate multiple small models to meet real-time and resource constraints. The typical scheme is as follows:

- **Query routing & Cascaded Inference:** It can send each request first to a compact model and escalate only hard cases to a larger model [121]. Additionally, confidence-based gating or ensemble voting among local LLMs enables distributed decision-making. For instance, in intelligent transportation, an RSU runs a small LLM, and if its prediction confidence is low, the query is forwarded to a more powerful LLM (possibly in the cloud). Such cascades dramatically reduce average latency and energy by keeping most computation on low-cost models. In real-world deployments, these ideas appear in

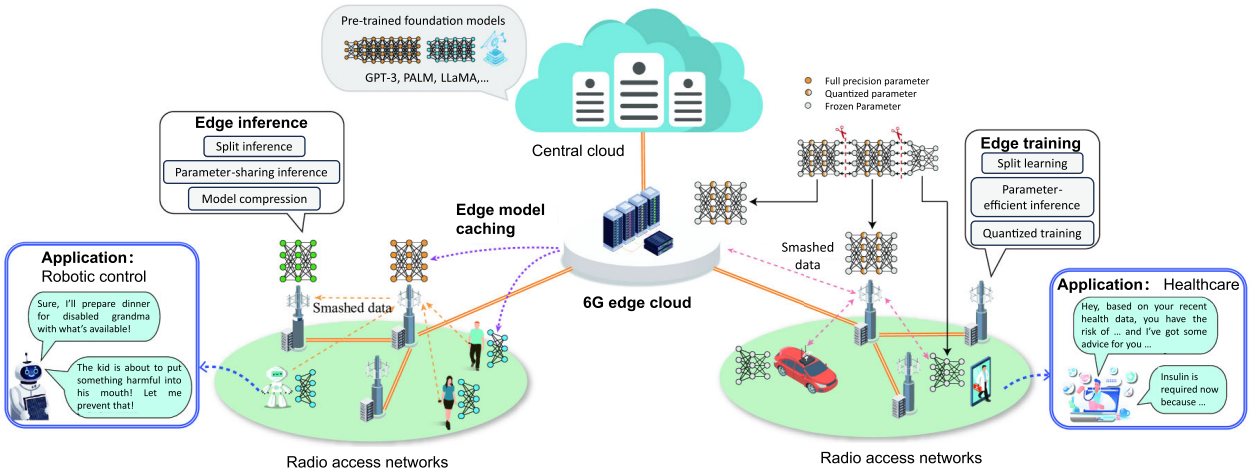

Fig. 5. LLM integrating split training in 6G Mobile Edge Computing (MEC) in [36]. This adapter co-training approach keeps user data on-device and drastically cuts communication, since only small adapter weights or token vectors are exchanged.

traffic forecasting and image-generation systems. For example, the LSGLLM-E architecture partitions a city's road network into subregions and deploys a lightweight spatio-temporal generative LLM on each RSU [122]. This edge-based LLM captures local spatio-temporal correlations and offloads work from the central cloud, achieving superior accuracy and efficiency. Similarly, a synergetic big-little framework co-trains a large cloud LLM with multiple small edge LLMs [19]. The cloud model provides high-level guidance while edge models capture local data, and a distributed training scheme aligns their parameters.

- **Cloud–Edge Co-Training:** On the training side, this scheme uses model-splitting and federated updates to share learning between devices. For example, split learning (SL) [123] can partition an LLM between edge and server so that only intermediate representations are transmitted. Combining SL with LoRA (Low-Rank Adaptation)-style adapters, multiple edge servers can jointly fine-tune a shared model in parallel. One study shows that integrating LoRA with parallel split federated learning allows large-model fine-tuning on edge GPUs in a reasonable time [36]. Fig. 5 shows the technical route of it when applied to the 6G edge.
- **Lightweight Inference:** At inference time, lightweight strategies like cascades, adapters, and token filtering further serve edge constraints. Confidence-based early exits give an edge to LLM output easy tokens immediately and offload only uncertain tokens for cloud processing [124]. Modular adapters enable a small model to be quickly customized to local context without invoking the full LLM. Even simple token pruning or split inference can cut latency. Placing the transformer encoders at the edge and only sending compact token embeddings to the cloud means that only tokens need to be exchanged.

Together, these techniques meet stringent edge requirements. They yield low-latency answers and reduced energy and communication costs by keeping most work on-device.

B. Dynamic Resource Orchestration

Edge-based LLM systems often employ early-exit and adaptive resource strategies to meet strict latency/energy targets on constrained devices [125]. For example, EdgeBERT uses an entropy-based early-exit policy to terminate inference at intermediate layers [120]. When confidence is high, it can significantly cut computation and latency. Similarly, CE-CoLLM [124] adopts a two-mode, cloud-edge pipeline. In a low-latency edge-only mode, it runs a lightweight early-exit LLM locally. While in a collaborative mode, high-confidence tokens are handled on-device and only low-confidence tokens (or residual computation) are offloaded to a larger cloud model. These approaches exemplify single-LLM orchestration. By combining early exits with adaptive scheduling and judicious model splitting between edge and cloud, they meet real-time constraints while saving energy and bandwidth [32], [126].

Distributed LLM inference extends these ideas across multiple devices or model replicas using advanced routing and scheduling policies.

- **Learned Routing Policy:** It assigns queries to different models based on difficulty. For instance, Hybrid LLM [127] trains a router to steer queries to a small or large model by predicting each query's hardness, dynamically trading off cost and accuracy. In a similar spirit, Agreement-Based Cascading (ABC) builds a cascade of increasingly powerful models and uses ensemble agreement at each stage [128]. If a small-model ensemble agrees confidently, inference stops locally. Otherwise, it cascades the query to a bigger model. ABC's data-dependent routing yields large savings in cloud traffic while preserving accuracy.
- **Edge-oriented Framework:** It also performs cooperative scheduling and load balancing across devices. EdgeShard, for example, formulates a joint device-selection and partition problem. It shards the LLM across heterogeneous edge nodes and uses a dynamic-programming algorithm to allocate layers to devices, so as to minimize latency and maximize throughput [84], as shown in Fig. 6.

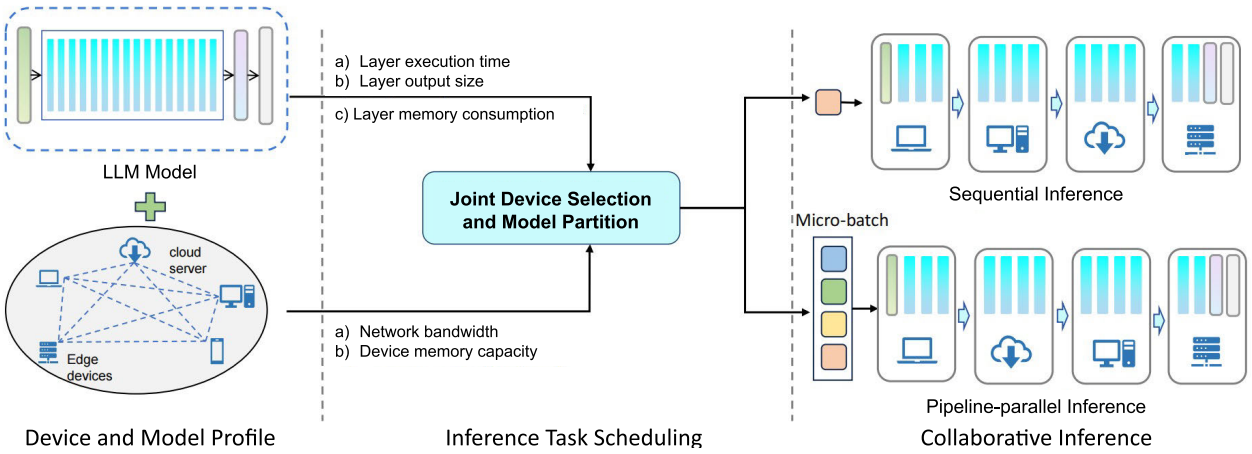

Fig. 6. The framework of EdgeLLM in [84]. It includes offline analysis, task scheduling optimization, and online collaborative LLM inference.

Likewise, Tian et al.'s synergetic big-cloud/small-edge architecture [19] trains and deploys a large cloud model alongside many tiny edge models. It uses a distributed task-oriented protocol, so that lightweight edge models handle common tasks locally. While the big model refines or assists only when needed, it harnesses collaborative intelligence to reduce latency and network load.

- **Multi-Agent Scheduling:** The MASITO framework [129] uses cooperating DRL (Deep Reinforcement Learning) agents on each edge server to schedule inference tasks and offloading under time/energy constraints. Then, it can effectively balance load and optimize accuracy across the network.

In summary, edge LLM orchestration now spans learned routers, cascaded ensembles, cooperative load distribution, and multi-agent DRL-based task allocation. All are designed balancing efficiency, accuracy, and latency in heterogeneous edge environments.

C. Model Context Protocol

The information transmission between LLMS and between LLMS and user cannot be done without the Model Context Protocol (MCP) [130]. Recent work on LLM context management focuses on compressing or streaming the input to fit edge constraints. For example, the In-Context Autoencoder (ICAE) trains a light-weight autoencoder to summarize a long input context into a small set of memory slots. It achieves roughly 4× context compression with minimal extra parameters [131]. This allows the LLM to condition on a much shorter representation of the full history, reducing GPU memory and latency. Similarly, EdgeInfinite [132] introduces a trainable memory-gating module within the Transformer. During inference, it keeps a few attention sink tokens and a sliding window of recent tokens in the KV (Key Value) cache, while compressing older tokens into a compact memory block. This gating mechanism is pretrained to preserve key semantic and positional information, enabling effective infinite-context inference on edge devices with little overhead.

Coordinating multiple LLMs on the edge introduces additional protocol complexity for sharing context among models. The specific methods are:

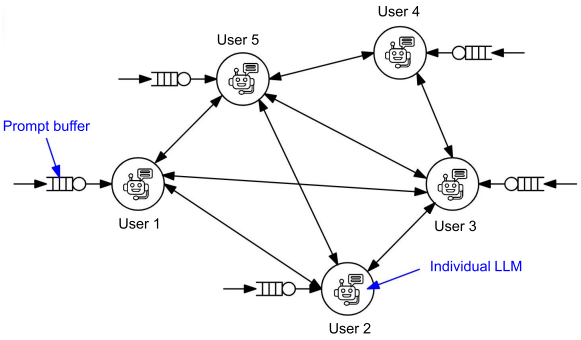

Fig. 7. Distributed MoA system model in [133]. Each device has its local prompt and makes inferences through the local LLM. Meanwhile, these prompts are sent to neighboring LLMs for reasoning, and then their responses are aggregated by one LLM.

- **Model-Distributed Inference:** One strategy is model-distributed inference, exemplified by MDI-LLM [134]. In this approach, a single-generation task is split across multiple devices. The LLM's layers are partitioned among nodes, which pipeline their computation to reduce idle time.
- **Separate Private and Public Memory Banks:** Another class of protocols treats memory by separating private vs. shared memory banks. For instance, the collaborative memory framework [135] proposes a two-tier memory architecture per LLM. Each model instance on an edge device maintains a private memory for its user-specific interactions and a shared memory for knowledge that can benefit others. When an LLM completes a query, it decides whether to write the interaction into shared memory or keep it private. Under limited edge resources, this selective sharing prevents unnecessary duplication while still enabling cross-LLM context reuse.
- **Retrieval-based Memory Fusion:** It is another related technique. For example, in the Memory Sharing (MS) framework [136], each agent logs its queries and answers as discrete memories. A communal memory store aggregates (prompt, response) pairs from many LLMs. When an LLM answers a new prompt, it performs retrieval over this shared pool to find relevant examples from other agents' experiences. These retrieved examples are then concatenated into the in-context prompt, effectively

fusing external context into the local inference. In [136], MS aids agents in identifying the most relevant examples for specific tasks by pooling memories across the network.

- **P2P Prompt Exchange Protocol:** This type of protocol represents the exchange of context among LLMs without a central server [28], [41]. In the Mixture-of-Agents (MoA) system [133], each device runs its LLM, and the devices intermittently communicate, as shown in Fig. 7. When a user generates a query, the device may broadcast that prompt to neighboring LLMs. Meanwhile, one LLM collects these multiple proposals and resolves them into a final answer. The P2P network ensures that updates propagate locally, but edge nodes have limited memory, the system must limit how many prompts sit in queues.
- **Agent-to-Agent (A2A) Protocol:** A core class of coordination mechanisms, the A2A protocol serves as the communication backbone for multi-LLM systems [137], enabling structured interaction and task collaboration between individual LLM agents. This protocol defines standardized rules for information exchange, task delegation, and result aggregation, thereby transforming a collection of LLMs into a unified, goal-oriented system.

D. Privacy Protection

In edge deployments of a single LLM, privacy protection often relies on minimizing data exposure and securing on-device computation. For example, data sanitization can strip personal identifiers from user inputs. As demonstrated by Rescriber [138], a system that helps users redact personal data from prompts and thereby reduces unnecessary disclosure. Local inference is also a key measure. By running the model entirely on the edge device, no raw user data needs to be sent to external servers, thereby keeping all personal inputs localized [139]. Additionally, advanced hardware and cryptographic safeguards are often employed. For instance, Trusted Execution Environments (TEE) encrypt and isolate model execution to protect data in use [140]. Finally, Differential Privacy (DP) techniques can be applied to model outputs or intermediate representations. Injecting calibrated noise into on-device embeddings can prevent user text from being exactly reconstructed downstream [141]. Together, these measures, including data minimization, local execution, encryption, and DP, form a layered defense that has become standard in edge LLM inference.

In edge-based multi-agent or multi-model settings, enhanced privacy measures are essential.

- **Federated Learning:** It has been widely proposed for LLMs to train or fine-tune shared models without exchanging raw data [142], [143]. Each device updates a local copy of the LLM, and only encrypted model deltas or gradients are shared, keeping user data on-device. However, even gradient updates can inadvertently leak information, so federated LLM frameworks typically layer on extra defenses. For instance, differential privacy or cryptographic masking can be applied to updates. They may be randomly perturbed or aggregated through secure multi-party computation so that individual contributions remain obscure [140].
- **Cryptographic Methods:** Likewise, this type of method allows LLMs to jointly infer without revealing inputs. One example is secure multi-party decoding, which confines user prompts to a TEE when collaborating with another LLM [144]. Emerging zero-knowledge proof techniques have also been explored to enable verification of LLM outputs without disclosing the inputs [145].
- **Role-based or Context-Aware Policies:** These policies can restrict cross-LLM data flow. For example, Shi et al. [146] introduced the Embedded Privacy-Enhancing Agents (EPEAgents). In this framework, each LLM might declare its role, and an embedded privacy LLM can filter communications so that only context-relevant information is shared with each model.

Furthermore, LLMs notoriously memorize training examples. Thus, adversarial prompts can trigger a model to regurgitate training data, undermining privacy protections [140]. These cross-LLM leakage and adversarial memorization attacks demonstrate that federated or encrypted LLMs still need robust DP, secure aggregation, and strict access controls to fully protect user data in multi-LLM edge deployments.

E. LLM Fine-Tuning

Edge computing provides a decentralized computing paradigm, bringing LLMs closer to data sources and users. In general, LLM fine-tuning for edge computing, the following aspects are mainly focused on. First, LoRA freezes the pre-trained weights of the LLM and introduces trainable low-rank matrices to adapt the model to specific edge tasks. For example, in some smart home applications, LoRA can be used to fine-tune LLMs on edge devices to better understand and respond to user voice commands [147]. Additionally, Quantization-aware Training (QAT) simulates the quantization process during fine-tuning to reduce the model's precision without significantly affecting its performance. For instance, the Edge-LLM framework adopts QAT to enable efficient LLM adaptation on edge devices [148]. The Edge-LLM framework also incorporates model pruning. By analyzing the sensitivity of different layers of the LLM to pruning, it dynamically allocates pruning sparsity for each layer. Then, it can effectively reduce model redundancy and improve computational efficiency.

In multi-LLM edge computing scenarios, LLM fine-tuning in this context has its unique aspects:

- **Federated Fine-Tuning:** Similar to federated learning, federated fine-tuning enables multiple edge devices to collaboratively fine-tune LLMs without directly sharing raw data [148]. Each device fine-tunes a local copy of the LLM based on its data and shares model updates with a central server or other devices. The server aggregates these updates to improve the global model. In [149], Zhang et al. proposed the Federated Instruction-Tuning (FedIT) framework, as show in Fig. 8 where the local training operations are performed at the client side, and scheduling and aggregation operations are performed at

the server side. Then, to protect data privacy, techniques like differential privacy and secure multi-party computation can be applied to the shared updates [150].

- **Split Learning for Fine-Tuning:** By segmenting the model and conducting collaborative training among multiple clients and servers, it can effectively reduce the computational and communication burden of a single client, thereby empowering the fine-tuning of the multi-LLM [151]. For instance, in the SplitLoRA framework [152], only activation and gradients need to be exchanged between the client and the server, rather than the entire LLM, which significantly reduces the communication bandwidth requirements. Furthermore, in edge computing, the network conditions of different devices vary greatly. The low communication overhead feature of SL makes it more suitable for heterogeneous networks, which can better meet the deployment requirements of LLM in different regions and under different network conditions.
- **Transfer Learning-based Fine-Tuning:** One LLM can be fine-tuned as a base model on a general edge task and then transferred to other related edge tasks for further fine-tuning [30]. This leverages the knowledge learned from the base model, reducing the amount of data and computational resources needed for fine-tuning on new tasks. For instance, an LLM fine-tuned on a general natural language processing task on edge devices can be transferred to specific tasks like edge-based smart customer service or smart education. It requires only minimal additional fine-tuning to achieve good performance.

In multi-LLM edge computing, fine-tuning faces challenges such as model compatibility, communication overhead, and data privacy protection. Ensuring the effective fine-tuning of multi-LLM while meeting the real-time and privacy requirements remains an active research area.

F. Multimodal Information Fusion

In edge computing, LLMs must process and fuse multimodal data to adapt to the complex environment of edge devices [153], [154]. In computer vision tasks like image description generation and visual question answering, LLMs can be combined with visual models. For instance, a pre-trained vision transformer extracts image features, which are then fused with text prompts input into the LLM to generate coherent and accurate textual descriptions. For instance, the LLM-Fusion model [155] leverages LLMs to integrate representations such as selfies, text descriptions, and molecular fingerprints for precise property prediction. Furthermore, LLM can also fuse multimodal sensor data to achieve various tasks. For example, You et al. [156] proposed a multimodal data fusion method based on LLM and attention mechanisms for traffic applications. It processes sensor and text data from vehicles and roads to improve traffic prediction accuracy. This technology will enable the scenarios described in Section III to yield benefits. For instance, in smart grid inspections, the images captured by the drones and the data collected by the sensors complement each other to construct a comprehensive assessment of the power facility status.

Moreover, in multi-LLM edge computing environments, multimodal information fusion exhibits different characteristics and advantages:

- **Collaborative Fusion of Multiple Models:** Multi-LLM systems can collaboratively fuse multimodal data. Each LLM can process specific modalities or modal combinations. The results are then fused and cross-validated to improve fusion accuracy and robustness. For example, one LLM can focus on processing visual data while another handles textual data. The fusion results from both models are combined for comprehensive scene understanding and decision-making [41].
- **Specialized Fusion Strategies:** Multi-LLM can adopt specialized fusion strategies based on the characteristics of different modalities and tasks. For instance, in multimodal recommendation systems, some LLMs can specialize in extracting user preferences from text data, while others analyze image or audio data to provide supplementary information. The fusion results help generate more personalized recommendations. A survey on multimodal recommendation systems in the LLM era highlights the advantages of integrating LLMs into multimodal recommenders, such as advanced preference summarization, context-aware fusion, and personalized content generation.
- **Attention-based Fusion:** This approach is highly effective in multimodal applications as it can handle noise and uncertainties in multimodal data. For example, Q-Former uses attention mechanisms to align multimodal features before generating the final output through LLM [157]. Then, researchers directly embedded the adapter into the LLM and allowed for end-to-end training that included alignment [158]. In a multi-LLM system, each LLM can process different modalities or modality combinations using its attention-based fusion module. The fused results are then combined to produce the final output. This method allows for in-depth exploration of the relationships between modalities and focuses on the important information in each modality.
- **Cross-modal Feature Alignment:** Techniques such as the Att-Sinkhorn [159] method combine the Sinkhorn metric with attention mechanisms to address the optimal transport problem between probability distributions of different modalities, thereby improving the accuracy of multimodal feature alignment. In multi-LLM systems, cross-modal feature alignment can be used to align multimodal features from different LLMs into a shared semantic space. This enables better collaboration and information sharing among LLMs, enhancing the overall performance of multimodal fusion.

Overall, multi-LLM provides unique advantages for multimodal information fusion. They are natural benefits brought by multiple perspectives.

G. Lessons Learned

According to the analysis of the above key technologies, achieving the synergy of multiple LLMs on the edge side needs to take the "performance-resource-security" triangular

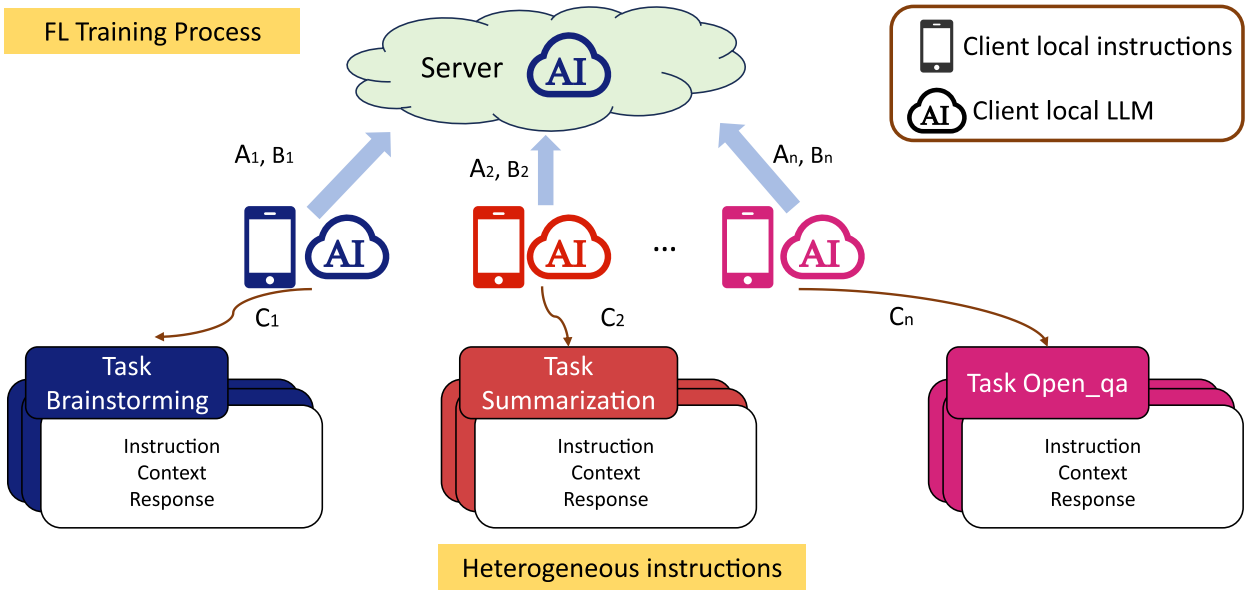

Fig. 8. FedIT framework in [149]. All the dense layers of each LLM are introduced into a parallel LoRA module. Thus, the number of trainable parameters is significantly reduced, thereby lowering the computational and communication overhead.

constraint as the core. Multimodal information fusion and model context protocols have enhanced the performance of EGI, but lightweight and resource orchestration are required to address the bottleneck of resource constraints. In addition, privacy protection and fine-tuning strategies are implemented to ensure the reliability and accuracy of the generated content. Therefore, in the future, it is necessary to focus on these three aspects to break through the triangle limitations.

V. Trustworthy Multi-LLM for Edge Computing

In this section, we emphasize the trustworthiness challenges of multi-LLM and design a blockchain-based approach as a tutorial. Subsequently, we also summarize the publicly available datasets related to multi-LLM, intending to contribute to the support of various EGI tasks.

A. A Case Study of Trustworthy Multi-LLM for EGI

In edge computing, devices are often in open wireless scenarios and are vulnerable to various attacks. Furthermore, edge devices are scattered, and the environment is complex, with high security threats. These factors will all weaken the response trustworthiness when multi-LLM empowers the edge. The term "trustworthiness" here primarily refers to the authenticity and reliability of the response, ensuring it can provide excellent services to users without being maliciously tampered with by attackers. Specifically, the potential factors influencing credibility in multi-LLM are:

- **Difficult to Determine the Response Trustworthiness:** Different LLMs use different corpora, training methods, and scenario orientations, resulting in differences in the output for the same problem. Although there are ways of collaboration, competition, and integration, it is difficult to determine which output has the best specific credibility and quality.
- **Threat of Malicious Behavior:** Once the deployed device is dishonest, the content generated by the LLM may generate misleading responses due to viruses, Trojans, or the operator's intentions. Furthermore, there are currently malicious models deliberately designed to actively deceive users and obtain illegal data privacy and economic benefits, such as WormGPT [160].
- **Defects of the Collaboration Framework:** Traditional Multi-LLM relies on the coordinated response of the central node, while it may be maliciously attacked or fail, thereby leading to the risk of Single-Point Failure (SPF) [161], [162]. Meanwhile, the centralized coordination approach also has bottlenecks in terms of efficiency [163]. Furthermore, the existing collaboration frameworks are unable to verify the credibility of LLM devices and responses, and malicious LLMs may contaminate the collaboration results [164].
- **Lack of Transparency and Traceability:** The interactive cooperation of multi-LLM, although it improves the response quality, leads to an untraceable source of the response due to the opacity of the collaboration mode. If the source of the generated content cannot be traced, it will not be easy to audit the reliability of the response [165], [166].

1) Blockchain-driven multi-LLM: Fortunately, as a decentralized ledger technology, blockchain can drive this multi-LLM system safely and efficiently, thereby providing a reliable optimization method for edge networks. On the one hand, blockchain consensus enables multi-LLMs to respond with the best quality without relying on trusted third parties, overcoming the SPF and efficiency bottlenecks of centralized coordination [167], [168], [169]. On the other hand, the immutability and traceability of blockchain ensure the credibility of the responses generated in the multi-LLM system [164], [170], [171]. Here, based on the work in [41], we introduce

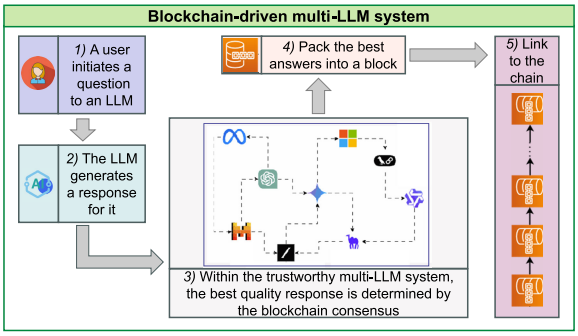

Fig. 9. The blockchain-driven trustworthy multi-LLM system in [41]. In a multi-LLM framework, every response generated by a specific LLM for a user undergoes verification and cross-comparison across all participating LLMs. This mechanism guarantees that the system delivers to users the most reliable answers.

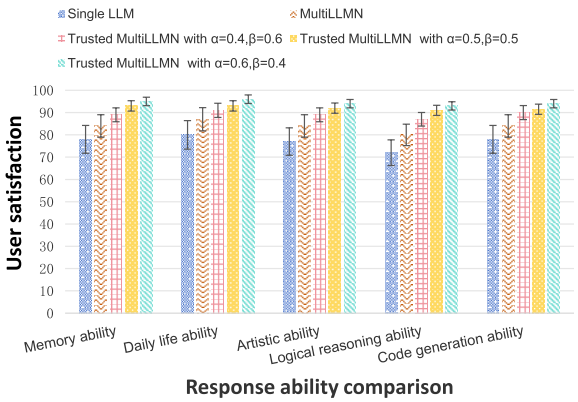

Fig. 10. The comparison of users' ratings of various LLM schemes in [28]. This result demonstrates the advantages of the multi-LLM system over a single LLM, and also reflects the importance of trustworthiness to the multi-LLM.

how blockchain drives multi-LLM collaboration. As shown in Fig. 9, the specific five steps of this method are as follows:

- **User Initiation:** An individual submits a query to a reliable multi-LLM system.
- **LLM Response Creation:** Every LLM within the system formulates an answer tailored to the user's query. Subsequently, these LLMs communicate with one another through broadcast protocols within the blockchain's P2P network.
- **Blockchain Consensus Mechanism:** Achieving consensus is crucial for identifying the best response among those generated by different LLMs. It implements a voting-based consensus to evaluate and choose the final output. The designated consensus nodes then transmit this result back to the user.
- **Block Formation:** The top-ranked response decided by the blockchain consensus process is encapsulated into a block. To safeguard the security, immutability, and traceability of the consensus outcome, the block incorporates the hash value of the optimal solution along with its corresponding timestamp.
- **Blockchain Extension:** Blocks holding the consensus results are appended to the blockchain and are stored in a decentralized fashion across smart devices that host the LLMs.

Furthermore, Luo et al. [28] designed a Weighted Byzantine Fault Tolerance (WBFT) consensus based on response quality and trust value. In this consensus, the voting rights of each LLM are jointly determined by its generated content ability and credibility. The proportions of these two in the voting weights are α and β respectively. This setting significantly enhances the voting rights of LLMs with high generation capabilities and credibility, and weakens the influence of malicious LLMs on the trustworthy multi-LLM system. The LLMs in this WBFT-driven trustworthy multi-LLM system are interconnected by Python mobilizing their interfaces. It runs on a high-performance server equipped with a 96-core Intel(R) Xeon(R) Gold 5220R CPU @ 2.20 GHz and 1 TB memory.

Then, the authors gathered 15 volunteers from all over the world to rate the generation capabilities of individual LLMs, multi-LLM without blockchain participation, and the trustworthy multi-LLM system driven by WBFT. When the values of α and β are different, the average score statistics of these schemes are shown in Fig. 10. This comparison result fully demonstrates the advantages of trustworthy multi-LLM driven by blockchain consensus in providing services for users, which is significantly superior to a single LLM and the multi-LLM system without blockchain participation. Also, the verification results of different values of α and β reflect the importance of LLM response quality for user satisfaction. Conversely, this work also reveals the crucial impact of LLM credibility on the WBFT consensus performance.

2) Trustworthy Multi-LLM for EGI: The multi-LLM system integrating blockchain can bring more benefits to the EGI scenarios. We take the four scenarios in Section III as examples to elaborate on them respectively.

- **Elderly Care:** The collaborative verification of medical advice by multiple LLMs can avoid incorrect decisions caused by the illusion or bias of a single model. On this basis, the immutability of blockchain ensures that sensitive health data is only recorded after being encrypted, meeting the compliance requirements for medical privacy. In addition, this system can also prevent the compromised wearable device LLM from sending fake alerts, such as false fall detection, ensuring the authenticity of emergency responses.
- **Smart Grid Inspection:** Multiple LLMs collaboratively analyze sensor data and drones images, and avoid missed detection by a single model through cross-validation. The participation in blockchain consensus requires the majority of LLMs to reach an agreement before triggering an alert, significantly reducing the rate of false alerts. Meanwhile, all inspection operations, such as equipment status updates and maintenance instructions, will be recorded on the chain. This will facilitate the tracking of responsible parties and prevent internal personnel or external hackers from tampering with the data.
- **Intelligent Transportation:** When intersection LLMs collaborate with vehicle LLMS to optimize traffic lights and path planning, blockchain consensus can prevent malicious nodes from forging traffic data, such as false congestion reports, and prevent traffic paralysis or

accidents. Furthermore, vehicle sensor data needs to be verified by multiple LLMS before being used for global scheduling to prevent invaded vehicles from sending misleading information.

- **LAENets:** When multiple LLMs negotiate flight paths in real time, blockchain consensus can prevent malicious drones from sending false position data and causing collisions. Meanwhile, the instruction sources and execution results of logistics or communication tasks are recorded on the chain, facilitating dispute traceability and efficiency optimization.

B. Open-Source Datasets

High-quality data plays an important role in the training and performance evaluation of multi-LLM. In this section, we outline the publicly available datasets related to multi-LLM. Table III lists the brief introductions of these datasets.

- **FSPO (Few-Shot Preference Optimization):** Singh et al. [172] designed FSPO.[8] It uses a very small amount of real user preference to optimize the LLMs and make its output more in line with the preferences of specific individual users.
- **BabbleBeaver[9]:** It provides some prompt examples for multiple LLMs, which can facilitate conversations among LLMs, including OpenAI, Google Gemini, Mistral, Anthropic, Cohere, Ollama, etc.
- **MMLU[10]:** This is a large-scale multi-task test dataset used to measure the model's ability to understand multi-domain knowledge and solve problems, aiming to assess the model's ability to understand and solve problems in a wide range of knowledge domains.
- **LLM-QA[11]:** It is used to evaluate the accuracy of multiple LLMs on multiple-choice tasks. By comparing their performances, identify the model that performs best on a specific task.
- **GSM8K[12]:** It is used for training and evaluating the abilities of different LLMs in solving primary-school mathematics. Specifically, it helps the model learn to identify its own mistakes and try repeatedly until the correct solution is found.
- **ChatGLM[13]:** The dataset mentioned by ChatGLM is a history-related dataset used for fine-tuning LLMs to build a Chinese chatbot. It provides some prompts in JSON format.

VI. Future Research Directions

Although the capabilities of multi-LLM in model compression, resource orchestration, model context protocol, privacy protection, LLM fine-tuning and multimodal information fusion are impressive, its application in edge computing is still in its early stages. This section aims to explore the research directions related to multi-LLM-enabled EGI.

[8]https://github.com/Asap7772/fewshot-preference-optimization
[9]https://github.com/open-build/BabbleBeaver/tree/main
[10]https://huggingface.co/datasets/cais/mmlu
[11]https://github.com/collectioncard/LLM-QA-Analysis/tree/main
[12]https://openai.com/index/solving-math-word-problems/
[13]https://github.com/SchweitzerGAO/awesome-chinese-chatbot

A. Lightweight Multimodal LLM

Current research has confirmed that lightweight multimodal models are of great significance for resource-constrained environments such as edge computing. For example, in the field of multimodal emotion recognition, existing studies have proposed a lightweight neural network architecture. It only uses approximately 2.7 M of parameters when analyzing multimodal information [173]. In the future, on this basis, model compression and optimization techniques such as quantization, pruning and knowledge distillation can be further explored to further reduce the model size and computational complexity, enabling it to run more efficiently on edge devices.

In terms of efficient architecture design, the design concepts of lightweight networks such as MobileNet [174] can be drawn upon to design structures such as depth-separable convolution, suitable for multimodal data processing. This method can reduce parameter redundancy and improve computational efficiency. In addition, it is necessary to study the hardware adaptation and acceleration technologies of multimodal models in view of the characteristics of edge computing. For instance, collaborate with chip manufacturers to optimize the execution efficiency of models on specific hardware platforms, or develop specialized edge AI chips to support the efficient operation of lightweight multimodal models.

B. Cross-Domain Generalization

Realizing cross-domain generalization capability is one of the key challenges faced by multi-LLM systems. Existing studies have begun to focus on how to improve the adaptability of the model in different fields. For example, through domain adaptation algorithms such as adversarial training [175], the feature differences between the source domain and the target domain can be reduced, and the generalization performance of the model in new domains can be improved. In the future, it is necessary to further develop more effective domain adaptation algorithms and explore how to achieve cross-domain knowledge transfer and integration in multi-LLM systems. By constructing knowledge graphs and other means to integrate semantic information from different fields [176], it provides the model with richer background knowledge and helps it make more accurate decisions in cross-domain tasks.

In addition, optimizing the pre-training and fine-tuning strategies is also an important direction for improving the cross-domain generalization ability [177]. Multi-domain data can be adopted for pre-training, enabling the model to learn broader general knowledge at the initial stage. In the fine-tuning stage, the method of transfer learning is adopted to make targeted adjustments to the model according to the characteristics of the target domain.

C. Trustworthy AI Governance

Improving the credibility of the multi-LLM system is the key to achieving its wide application [178]. In terms of model transparency and interpretability, it is necessary to study in the future how to provide a reasonable explanatory basis for the output of the model. For example, develop an interpretation

TABLE III
OPEN-SOURCE DATASET RELATED TO MULTI-LLM

Dataset	Description	Composition
FSPO	A dataset for personalizing different LLMs	Contain a dataset with preferences
BabbleBeaver	A prompt word dataset for multiple LLMs	Provide 50 typical prompt words
MMLU	A dataset for testing the performance of different LLMs	Include 15,908 questions from 57 disciplines
LLM-QA	A dataset for testing the selection accuracy of multiple LLMs	Contain 30 question-answer pairs
GSM8K	A primary school mathematics dataset for evaluating multiple LLMs	Contain 8,500 question-answer pairs
ChatGLM	A historically relevant dataset for pre-training multiple LLMs	Provide some prompts in JSON format

generator and adopt technologies such as attention mechanism visualization to demonstrate the concerns and reasoning paths of the model when dealing with multimodal data.

Enhanced security and robustness are also important components of trusted AI governance. The security design of the multi-LLM system needs to be further strengthened, and techniques such as adversarial training and data augmentation should be adopted to improve the robustness of the model [179]. Meanwhile, study the vulnerability detection and repair methods of the model to promptly identify and solve potential security issues. In terms of privacy protection mechanisms, advanced cryptographic technologies are adopted to achieve collaborative processing of multimodal data without disclosing data privacy, preventing data leakage and abuse [180]. Furthermore, accountability for the multi-LLM system is also very important. When multiple LLMs work together, if they provide incorrect judgments for edge users, it could lead to potential losses. One promising approach is to use blockchain to record the process of collaboration, debate, and integration among LLMs, so that it can be audited later.

D. Multi-LLM Reasoning

Reasoning methods are crucial for the multi-LLM system because they significantly enhance its ability to handle complex tasks and improve the reliability of decision-making. The multi-LLM system combined with Retrieval Enhanced Generation (RAG) and Retrieval Enhanced Perception (RAP) can handle multimodal data more efficiently and improve the accuracy and relevance of the generated content [181], [182]. For example, when dealing with complex image description tasks, the multi-LLM system can generate more detailed descriptions by retrieving relevant image and text information. In the future, the collaborative mechanism of multiple LLMs in RAG and RAP can be explored to optimize the retrieval strategy and enhance the complementarity among models.

Additionally, Chain of Thought (CoT) can enhance the reasoning ability in complex tasks by simulating the human step-by-step reasoning process in multi-LLM systems [183]. For example, multi-LLM systems can achieve more detailed reasoning steps by decomposing problems and assigning them to different expert models. Meanwhile, the world model, as a structured framework integrating the laws and physical rules of the real world, can eliminate cognitive conflicts among LLMs and bring core advantages such as cognitive consistency and task collaboration to the multi-LLM system [184]. Furthermore, agentic AI can simulate the autonomous decision-making and interaction of agents in multi-LLM systems [185], [186], achieving more efficient information sharing and collaborative work to handle complex dynamic tasks.

VII. CONCLUSION

This survey comprehensively explores multi-LLM systems in edge computing, with a focus on their evolution from traditional edge artificial intelligence models to individual LLMS and then to multi-LLM systems. It discusses the typical application scenarios of this example, thereby leading to key supporting technologies such as model compression and dynamic data. Meanwhile, the survey emphasizes that multi-LLM needs to make robust decisions and provide reliable responses in an environment with high reliability and privacy. After providing relevant open-source data, the survey also discussed future research directions, including lightweight architecture, trusted governance and other issues. Overall, multi-LLM systems have demonstrated great potential in promoting the development of edge computing towards more intelligent and autonomous applications, effectively meeting the demands of our complex digital world.

REFERENCES

[1] L. Song, G. Sun, H. Yu, and D. Niyato, "ESPD-LP: Edge service predeployment based on location prediction in MEC," *IEEE Trans. Mobile Comput.*, vol. 24, no. 6, pp. 5551–5568, Jun. 2025.
[2] H. Chen et al., "Toward edge general intelligence via large language models: Opportunities and challenges," *IEEE Netw.*, vol. 39, no. 5, pp. 263–271, Sep. 2025.
[3] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020.
[4] M. A. Ferrag et al., "Generative AI in cybersecurity: A comprehensive review of LLM applications and vulnerabilities," *Internet Things Cyber-Phys. Syst.*, vol. 5, pp. 1–46, Mar. 2025.
[5] C. Liang et al., "Generative AI-driven semantic communication networks: Architecture, technologies, and applications," *IEEE Trans. Cognit. Commun. Netw.*, vol. 11, no. 1, pp. 27–47, Feb. 2025.
[6] Z. Guo, F. Tang, L. Luo, M. Zhao, and N. Kato, "A survey on applications of large language model-driven digital twins for intelligent network optimization," *IEEE Commun. Surveys Tuts.*, early access, May 9, 2025, doi: 10.1109/COMST.2025.3568637.
[7] Y. Chang et al., "A survey on evaluation of large language models," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, pp. 1–45, 2024.
[8] C. Zhao et al., "Generative AI for secure physical layer communications: A survey," *IEEE Trans. Cognit. Commun. Netw.*, vol. 11, no. 1, pp. 3–26, Feb. 2025.
[9] R. Zhang et al., "Generative AI-enabled vehicular networks: Fundamentals, framework, and case study," *IEEE Netw.*, vol. 38, no. 4, pp. 259–267, Jul. 2024.
[10] R. Zhang et al., "Toward edge general intelligence with agentic AI and agentification: Concepts, technologies, and future directions," 2025, *arXiv:2508.18725*.

[11] L. He, L. Fan, X. Lei, P. Fan, A. Nallanathan, and G. K. Karagiannidis, "The road toward general edge intelligence: Standing on the shoulders of foundation models," *IEEE Commun. Mag.*, vol. 63, no. 9, pp. 164–170, Sep. 2025.

[12] R. Zhang et al., "Embodied AI-enhanced vehicular networks: An integrated vision language models and reinforcement learning method," *IEEE Trans. Mobile Comput.*, early access, Jun. 24, 2025, doi: 10.1109/TMC.2025.3582864.

[13] Z. Wang, Y. Shi, and K. B. Letaief, "Edge large AI models: Collaborative deployment and IoT applications," 2025, *arXiv:2505.03139*.

[14] R. Zhang et al., "Toward democratized generative AI in next-generation mobile edge networks," *IEEE Netw.*, early access, Feb. 11, 2025, doi: 10.1109/MNET.2025.3541078.

[15] X. Ma, G. Fang, and X. Wang, "LLM-pruner: On the structural pruning of large language models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 21702–21720.

[16] R. Agrawal, H. Kumar, and S. R. Lnu, "Efficient LLMs for edge devices: Pruning, quantization, and distillation techniques," in *Proc. Int. Conf. Mach. Learn. Auto. Syst. (ICMLAS)*, Mar. 2025, pp. 1413–1418.

[17] Z. Li, T. Li, W. Feng, M. Guizani, and H. Yu, "PRIMA.CPP: Speeding up 70B-scale LLM inference on low-resource everyday home clusters," 2025, *arXiv:2504.08791*.

[18] Z. Li, W. Feng, M. Guizani, and H. Yu, "TPI-LLM: Serving 70B-scale LLMs efficiently on low-resource edge devices," 2024, *arXiv:2410.00531*.

[19] Y. Tian et al., "An edge-cloud collaboration framework for generative AI service provision with synergetic big cloud model and small edge models," *IEEE Netw.*, vol. 38, no. 5, pp. 37–46, Sep. 2024.

[20] F. Wan, X. Huang, D. Cai, X. Quan, W. Bi, and S. Shi, "Knowledge fusion of large language models," 2024, *arXiv:2401.10491*.

[21] D. M. Owens et al., "A multi-LLM debiasing framework," 2024, *arXiv:2409.13884*.

[22] M. Ding et al., "Easy2Hard-bench: Standardized difficulty labels for profiling LLM performance and generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 37, 2024, pp. 44323–44365.

[23] X. Wang et al., "Wireless hallucination in generative AI-enabled communications: Concepts, issues, and solutions," 2025, *arXiv:2503.06149*.

[24] S. Feng, W. Shi, Y. Wang, W. Ding, V. Balachandran, and Y. Tsvetkov, "Don't hallucinate, abstain: Identifying LLM knowledge gaps via multi-LLM collaboration," 2024, *arXiv:2402.00367*.

[25] J. Lu, Z. Pang, M. Xiao, Y. Zhu, R. Xia, and J. Zhang, "Merge, ensemble, and cooperate! A survey on collaborative strategies in the era of large language models," 2024, *arXiv:2407.06089*.

[26] S. Z. Shen, H. Lang, B. Wang, Y. Kim, and D. Sontag, "Learning to decode collaboratively with multiple language models," 2024, *arXiv:2403.03870*.

[27] S. Wang, J. Deng, Q. Li, J. Wu, and Z. Zhao, "Performance analysis on the applications of large language models: A case for elderly care," in *Proc. IEEE Int. Conf. High Perform. Comput. Commun. (HPCC)*, Dec. 2024, pp. 145–151.

[28] H. Luo et al., "A weighted Byzantine fault tolerance consensus driven trusted multiple large language models network," 2025, *arXiv:2505.05103*.

[29] Y. Huang, "Levels of AI agents: From rules to large language models," 2024, *arXiv:2405.06643*.

[30] Y. Dong, H. Zhang, C. Li, S. Guo, V. C. M. Leung, and X. Hu, "Fine-tuning and deploying large language models over edges: Issues and approaches," 2024, *arXiv:2408.10691*.

[31] S. Bhardwaj, P. Singh, and M. K. Pandit, "A survey on the integration and optimization of large language models in edge computing environments," in *Proc. 16th Int. Conf. Comput. Autom. Eng. (ICCAE)*, Mar. 2024, pp. 168–172.

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

[33] R. Wang, Z. Gao, L. Zhang, S. Yue, and Z. Gao, "Empowering large language models to edge intelligence: A survey of edge efficient LLMs and techniques," *Comput. Sci. Rev.*, vol. 57, Aug. 2025, Art. no. 100755.

[34] G. Qu, Q. Chen, W. Wei, Z. Lin, X. Chen, and K. Huang, "Mobile edge intelligence for large language models: A contemporary survey," *IEEE Commun. Surveys Tuts.*, early access, Jan. 9, 2025, doi: 10.1109/COMST.2025.3527641.

[35] S. Hadish, V. Bojković, M. Aloqaily, and M. Guizani, "Language models at the edge: A survey on techniques, challenges, and applications," in *Proc. 2nd Int. Conf. Found. Large Lang. Models (FLLM)*, Nov. 2024, pp. 262–271.

[36] 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," 2023, *arXiv:2309.16739*.

[37] C. Zhao et al., "Edge general intelligence through world models and agentic AI: Fundamentals, solutions, and challenges," 2025, *arXiv:2508.09561*.

[38] Z. Chen et al., "Harnessing multiple large language models: A survey on LLM ensemble," 2025, *arXiv:2502.18036*.

[39] S. Feng et al., "When one LLM drools, multi-LLM collaboration rules," 2025, *arXiv:2502.04506*.

[40] L. Chen and G. Varoquaux, "What is the role of small models in the LLM era: A survey," 2024, *arXiv:2409.06857*.

[41] H. Luo et al., "A trustworthy multi-LLM network: Challenges, solutions, and a use case," 2025, *arXiv:2505.03196*.

[42] A. P. Behera, J. P. Champati, R. Morabito, S. Tarkoma, and J. Gross, "Towards efficient multi-LLM inference: Characterization and analysis of LLM routing and hierarchical techniques," 2025, *arXiv:2506.06579*.

[43] X. Dai, J. Li, X. Liu, A. Yu, and J. C. S. Lui, "Cost-effective online multi-LLM selection with versatile reward models," 2024, *arXiv:2405.16587*.

[44] C. Nandkumar and L. Peternel, "Enhancing supermarket robot interaction: An equitable multi-level LLM conversational interface for handling diverse customer intents," *Frontiers Robot. AI*, vol. 12, Apr. 2025, Art. no. 1576348.

[45] M. Yuan, J. Chen, Z. Xing, G. Mohammadi, and A. Quigley, "A case study of scalable content annotation using multi-LLM consensus and human review," 2025, *arXiv:2503.17620*.

[46] V. Mahadevan, S. Zhang, and R. Chandra, "GameChat: Multi-LLM dialogue for safe, agile, and socially optimal multi-agent navigation in constrained environments," 2025, *arXiv:2503.12333*.

[47] S. Feng et al., "Modular pluralism: Pluralistic alignment via multi-LLM collaboration," 2024, *arXiv:2406.15951*.

[48] E. Y. Chang, "SocraSynth: Multi-LLM reasoning with conditional statistics," 2024, *arXiv:2402.06634*.

[49] A. Estornell and Y. Liu, "Multi-llm debate: Framework, principals, and interventions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 37, 2024, pp. 28938–28964.

[50] J. Fang et al., "Multi-LLM text summarization," 2024, *arXiv:2412.15487*.

[51] D. Mao, D. Zhang, A. Zhang, and Z. Zhao, "MLSDET: Multi-LLM statistical deep ensemble for Chinese AI-generated text detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2025, pp. 1–5.

[52] J. Fang, Y. Shen, Y. Wang, and L. Chen, "Improving the end-to-end efficiency of offline inference for multi-LLM applications based on sampling and simulation," 2025, *arXiv:2503.16893*.

[53] L. Liu, D. Zhang, S. Li, G. Zhou, and E. Cambria, "Two heads are better than one: Zero-shot cognitive reasoning via multi-LLM knowledge fusion," in *Proc. 33rd ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2024, pp. 1462–1472.

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

[55] Y. Yang, Y. Ma, H. Feng, Y. Cheng, and Z. Han, "Minimizing hallucinations and communication costs: Adversarial debate and voting mechanisms in LLM-based multi-agents," *Appl. Sci.*, vol. 15, no. 7, p. 3676, Mar. 2025.

[56] C. Bandi and A. Harrasse, "Adversarial multi-agent evaluation of large language models through iterative debates," 2024, *arXiv:2410.04663*.

[57] C.-Y. Hsu et al., "Prediction of methane hydrate equilibrium in saline water solutions based on support vector machine and decision tree techniques," *Sci. Rep.*, vol. 15, no. 1, p. 11723, Apr. 2025.

[58] D. L. T. Wong, Y. Li, D. John, W. K. Ho, and C.-H. Heng, "An energy efficient ECG ventricular ectopic beat classifier using binarized CNN for edge AI devices," *IEEE Trans. Biomed. Circuits Syst.*, vol. 16, no. 2, pp. 222–232, Apr. 2022.

[59] A. A. Ahmed et al., "Secure AI for 6G mobile devices: Deep learning optimization against side-channel attacks," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3951–3959, Feb. 2024.

[60] J. Curzon, T. A. Kosa, R. Akalu, and K. El-Khatib, "Privacy and artificial intelligence," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 96–108, Apr. 2021.

[61] D. Zha et al., "Data-centric artificial intelligence: A survey," *ACM Comput. Surv.*, vol. 57, no. 5, pp. 1–42, 2023.

[62] S. K. Radha and O. Goktas, "On the reasoning capacity of AI models and how to quantify it," 2025, *arXiv:2501.13833*.

[63] M. Steidl, M. Felderer, and R. Ramler, "The pipeline for the continuous development of artificial intelligence models—Current state of research and practice," *J. Syst. Softw.*, vol. 199, Jan. 2023, Art. no. 111615.

[64] R. Singh and S. S. Gill, "Edge AI: A survey," *Internet Things Cyber-Phys. Syst.*, vol. 3, pp. 71–92, Apr. 2023.

[65] X. Ma et al., "Megalodon: Efficient LLM pretraining and inference with unlimited context length," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 37, 2024, pp. 71831–71854.

[66] Y. Huang et al., "Large language models for networking: Applications, enabling techniques, and challenges," *IEEE Netw.*, vol. 39, no. 1, pp. 235–242, Jan. 2025.

[67] S. Kou, C. Yang, and M. Gurusamy, "GIA: LLM-enabled generative intent abstraction to enhance adaptability for intent-driven networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 11, no. 2, pp. 999–1012, Apr. 2025.

[68] R. Zhang et al., "Generative AI for space-air-ground integrated networks," *IEEE Wireless Commun.*, vol. 38, no. 4, pp. 10–20, 2024.

[69] J. Wang, "A tutorial on LLM reasoning: Relevant methods behind ChatGPT o1," 2025, *arXiv:2502.10867*.

[70] H. Zhou et al., "Large language model (LLM) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 3, pp. 1955–2005, 3rd Quart., 2025.

[71] X. Zhang et al., "Beyond the cloud: Edge inference for generative large language models in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 24, no. 1, pp. 643–658, Jan. 2025.

[72] R. Zhang et al., "Generative AI agents with large language model for satellite networks via a mixture of experts transmission," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 12, pp. 3581–3596, Dec. 2024.

[73] Y. Cao et al., "Toward generalizable evaluation in the LLM era: A survey beyond benchmarks," 2025, *arXiv:2504.18838*.

[74] S. Xiao et al., "Distill-VQ: Learning retrieval oriented vector quantization by distilling knowledge from dense embeddings," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 1513–1523.

[75] C. Deng, X. Fang, X. Wang, and K. Law, "Software orchestrated and hardware accelerated artificial intelligence: Toward low latency edge computing," *IEEE Wireless Commun.*, vol. 29, no. 4, pp. 110–117, Aug. 2022.

[76] P. Patel et al., "Splitwise: Efficient generative LLM inference using phase splitting," in *Proc. ACM/IEEE 51st Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2024, pp. 118–132.

[77] H. Zhou, Z. Feng, Z. Zhu, J. Qian, and K. Mao, "UniBias: Unveiling and mitigating LLM bias through internal attention and FFN manipulation," 2024, *arXiv:2405.20612*.

[78] L. Che, T. Q. Liu, J. Jia, W. Qin, R. Tang, and V. Pavlovic, "EAZY: Eliminating hallucinations in LVLMs by zeroing out hallucinatory image tokens," 2025, *arXiv:2503.07772*.

[79] A. Stephan, D. Zhu, M. Aßenmacher, X. Shen, and B. Roth, "From calculation to adjudication: Examining LLM judges on mathematical reasoning tasks," 2024, *arXiv:2409.04168*.

[80] L. Peng, Y. Zhang, and J. Shang, "Generating efficient training data via LLM-based attribute manipulation," 2023, *arXiv:2307.07099*.

[81] S. Raza, R. Sapkota, M. Karkee, and C. Emmanouilidis, "TRiSM for agentic AI: A review of trust, risk, and security management in LLM-based agentic multi-agent systems," 2025, *arXiv:2506.04133*.

[82] W. Feng et al., "Learning in chaos: Efficient autoscaling and self-healing for distributed training at the edge," 2025, *arXiv:2505.12815*.

[83] S. Marro et al., "A scalable communication protocol for networks of large language models," 2024, *arXiv:2410.11905*.

[84] M. Zhang, J. Cao, X. Shen, and Z. Cui, "EdgeShard: Efficient LLM inference via collaborative edge computing," 2024, *arXiv:2405.14371*.

[85] K. Rao, G. Coviello, P. Benedetti, C. Giuseppe De Vita, G. Mellone, and S. Chakradhar, "ECO-LLM: LLM-based edge cloud optimization," in *Proc. Workshop AI For Syst.*, Jun. 2024, pp. 7–12.

[86] I. Ferri-Molla, J. Linares-Pellicer, C. Aliaga-Torro, and J. Izquierdo-Domenech, "Multi-agent AI system for adaptive cognitive training in elderly care," in *Proc. 17th Int. Conf. Agents Artif. Intell.*, 2025, pp. 937–947.

[87] J.-H. Syu, J. C.-W. Lin, G. Srivastava, and K. Yu, "A comprehensive survey on artificial intelligence empowered edge computing on consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 1023–1034, Nov. 2023.

[88] M. A. Rahman and M. S. Hossain, "An Internet-of-Medical-Things-enabled edge computing framework for tackling COVID-19," *IEEE Internet Things J.*, vol. 8, no. 21, pp. 15847–15854, Nov. 2021.

[89] T. Shaik, X. Tao, L. Li, H. Xie, and J. D. Velásquez, "A survey of multimodal information fusion for smart healthcare: Mapping the journey from data to wisdom," *Inf. Fusion*, vol. 102, Feb. 2024, Art. no. 102040.

[90] Z. Shen, F. Ding, Y. Yao, A. Bhardwaj, Z. Guo, and K. Yu, "A privacy-preserving social computing framework for health management using federated learning," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 4, pp. 1666–1678, Aug. 2023.

[91] D. Xu et al., "LLMCad: Fast and scalable on-device large language model inference," 2023, *arXiv:2309.04255*.

[92] A. Zaboli, S. L. Choi, T.-J. Song, and J. Hong, "ChatGPT and other large language models for cybersecurity of smart grid applications," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Jul. 2024, pp. 1–5.

[93] H. Shi et al., "Review of the opportunities and challenges to accelerate mass-scale application of smart grids with large-language models," *IET Smart Grid*, vol. 7, no. 6, pp. 737–759, Dec. 2024.

[94] S. Madani, A. Tavasoli, Z. K. Astaneh, and P.-O. Pineau, "Large language models integration in smart grids," 2025, *arXiv:2504.09059*.

[95] H. Zhou, Z. Zhang, D. Li, and Z. Su, "Joint optimization of computing offloading and service caching in edge computing-based smart grid," *IEEE Trans. Cloud Comput.*, vol. 11, no. 2, pp. 1122–1132, Apr. 2023.

[96] D. Carrillo et al., "Boosting 5G on smart grid communication: A smart RAN slicing approach," *IEEE Wireless Commun.*, vol. 30, no. 5, pp. 170–178, Oct. 2023.

[97] C. Ling et al., "Domain specialization as the key to make large language models disruptive: A comprehensive survey," 2023, *arXiv:2305.18703*.

[98] A. Ali, A. U. Sahin, Ö. Özkasap, and Y.-D. Lin, "The universal fog proxy: A third-party authentication solution for federated fog systems with multiple protocols," *IEEE Netw.*, vol. 35, no. 6, pp. 285–291, Nov. 2021.

[99] J. Yan, T. Chen, Y. Sun, Z. Nan, S. Zhou, and Z. Niu, "Dynamic scheduling for vehicle-to-vehicle communications enhanced federated learning," *IEEE Trans. Wireless Commun.*, early access, Jun. 2, 2025, doi: 10.1109/TWC.2025.3573048.

[100] H. Xu et al., "GenAI-powered multi-agent paradigm for smart urban mobility: Opportunities and challenges for integrating large language models (LLMs) and retrieval-augmented generation (RAG) with intelligent transportation systems," 2024, *arXiv:2409.00494*.

[101] D. Mahmud et al., "Integrating LLMs with ITS: Recent advances, potentials, challenges, and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 5, pp. 5674–5709, May 2025.

[102] H. Quan, Q. Zhang, and J. Zhao, "Federated learning assisted intelligent IoV mobile edge computing," *IEEE Trans. Green Commun. Netw.*, vol. 9, no. 1, pp. 228–241, Mar. 2025.

[103] Q. Jiang et al., "Potential game based distributed IoV service offloading with graph attention networks in mobile edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 10912–10925, Sep. 2024.

[104] H. Luo et al., "ESIA: An efficient and stable identity authentication for Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5602–5615, Apr. 2024.

[105] X. Xu et al., "Multi-agent reinforcement learning based edge content caching for connected autonomous vehicles in IoV," *ACM Trans. Auto. Adapt. Syst.*, vol. 20, no. 3, pp. 1–26, Sep. 2025.

[106] M. Chen, C. Wang, X. He, F. Zhu, L. Wang, and A. V. Vasilakos, "Embodied artificial intelligence-enabled Internet of Vehicles: Challenges and solutions," *IEEE Veh. Technol. Mag.*, vol. 20, no. 2, pp. 63–70, Jun. 2025.

[107] L. Cai et al., "Secure physical layer communications for low-altitude economy networking: A survey," 2025, *arXiv:2504.09153*.

[108] Y. Wang et al., "Toward realization of low-altitude economy networks: Core architecture, integrated technologies, and future directions," 2025, *arXiv:2504.21583*.

[109] C. Zhao et al., "Generative AI-enabled wireless communications for robust low-altitude economy networking," 2025, *arXiv:2502.18118*.

[110] F. Zhu, F. Huang, Y. Yu, G. Liu, and T. Huang, "Task offloading with LLM-enhanced multi-agent reinforcement learning in UAV-assisted edge computing," *Sensors*, vol. 25, no. 1, p. 175, Dec. 2024.

[111] L. Cai et al., "Large language model-enhanced reinforcement learning for low-altitude economy networking," 2025, *arXiv:2505.21045*.

[112] P. McEnroe, S. Wang, and M. Liyanage, "A survey on the convergence of edge computing and AI for UAVs: Opportunities and challenges," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15435–15459, Sep. 2022.

[113] H. Luo, Y. Wu, G. Sun, H. Yu, and M. Guizani, "ESCM: An efficient and secure communication mechanism for UAV networks," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 3, pp. 3124–3139, Jun. 2024.

[114] D. Luo, Q. Cai, G. Sun, and H. Yu, "Split-chain based efficient blockchain-assisted cross-domain authentication for IoT," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 3, pp. 3209–3223, Jun. 2024.

[115] C. Zhao et al., "Temporal spectrum cartography in low-altitude economy networks: A generative AI framework with multi-agent learning," 2025, *arXiv:2505.15571*.

[116] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

[117] X. Jiao et al., "TinyBERT: Distilling BERT for natural language understanding," 2019, *arXiv:1909.10351*.

[118] M. Hussain et al., "Low-resource MobileBERT for emotion recognition in imbalanced text datasets mitigating challenges with limited resources," *PLoS ONE*, vol. 20, no. 1, Jan. 2025, Art. no. e0312867.

[119] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.

[120] T. Tambe et al., "EdgeBERT: Sentence-level energy optimizations for latency-aware multi-task NLP inference," in *Proc. 54th Annu. IEEE/ACM Int. Symp. Microarchit.*, Oct. 2021, pp. 830–844.

[121] M. J. Zellinger, R. Liu, and M. Thomson, "Cost-saving LLM cascades with early abstention," 2025, *arXiv:2502.09054*.

[122] Y. Rong, Y. Mao, X. He, and M. Chen, "Large-scale traffic flow forecast with lightweight LLM in edge intelligence," *IEEE Internet Things Mag.*, vol. 8, no. 1, pp. 12–18, Jan. 2025.

[123] Z. Lin, G. Qu, X. Chen, and K. Huang, "Split learning in 6G edge networks," *IEEE Wireless Commun.*, vol. 31, no. 4, pp. 170–176, Aug. 2024.

[124] H. Jin and Y. Wu, "CE-CoLLM: Efficient and adaptive large language models through cloud-edge collaboration," 2024, *arXiv:2411.02829*.

[125] Y. Zhang, L. Luo, G. Sun, H. Yu, and B. Li, "Deadline-aware online job scheduling for distributed training in heterogeneous clusters," *IEEE Trans. Cloud Comput.*, vol. 13, no. 2, pp. 590–604, Apr. 2025.

[126] M. Li, J. Gao, C. Zhou, X. S. Shen, and W. Zhuang, "Slicing-based artificial intelligence service provisioning on the network edge: Balancing AI service performance and resource consumption of data management," *IEEE Veh. Technol. Mag.*, vol. 16, no. 4, pp. 16–26, Dec. 2021.

[127] D. Ding et al., "Hybrid LLM: Cost-efficient and quality-aware query routing," 2024, *arXiv:2404.14618*.

[128] S. Kolawole, D. Dennis, A. Talwalkar, and V. Smith, "Agreement-based cascading for efficient inference," 2024, *arXiv:2407.02348*.

[129] A. B. Sada, A. Khelloufi, A. Naouri, H. Ning, N. Aung, and S. Dhelim, "Multi-agent deep reinforcement learning-based inference task scheduling and offloading for maximum inference accuracy under time and energy constraints," *Electronics*, vol. 13, no. 13, p. 2580, Jun. 2024.

[130] P. P. Ray, "A survey on model context protocol: Architecture, state-of-the-art, challenges and future directions," *TechRxiv*, 2025, doi: 10.36227/techrxiv.174495492.22752319/v1.

[131] T. Ge, J. Hu, X. Wang, S. Chen, and F. Wei, "In-context autoencoder for context compression in a large language model," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023, pp. 1–8.

[132] J. Chen et al., "EdgeInfinite: A memory-efficient infinite-context transformer for edge devices," 2025, *arXiv:2503.22196*.

[133] P. Mitra, P. Kaswan, and S. Ulukus, "Distributed mixture-of-agents for edge inference with large language models," 2024, *arXiv:2412.21200*.

[134] D. Macario, H. Seferoglu, and E. Koyuncu, "Model-distributed inference for large language models at the edge," 2025, *arXiv:2505.18164*.

[135] A. Rezazadeh, Z. Li, A. Lou, Y. Zhao, W. Wei, and Y. Bao, "Collaborative memory: Multi-user memory sharing in LLM agents with dynamic access control," 2025, *arXiv:2505.18279*.

[136] H. Gao and Y. Zhang, "Memory sharing for large language model based agents," 2024, *arXiv:2404.09982*.

[137] B. Yan et al., "Beyond self-talk: A communication-centric survey of LLM-based multi-agent systems," 2025, *arXiv:2502.14321*.

[138] J. Zhou, E. Xu, Y. Wu, and T. Li, "Rescriber: Smaller-LLM-powered user-led data minimization for LLM-based chatbots," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2025, pp. 1–28.

[139] B. Yan et al., "On protecting the data privacy of large language models (LLMs): A survey," 2024, *arXiv:2403.05156*.

[140] D. Li, Z. Zhang, M. Yao, Y. Cai, Y. Guo, and X. Chen, "TEESlice : Protecting sensitive neural network models in trusted execution environments when attackers have pre-trained models," *ACM Trans. Softw. Eng. Methodology*, vol. 34, no. 6, pp. 1–49, Jul. 2025.

[141] P. Mai, R. Yan, Z. Huang, Y. Yang, and Y. Pang, "Split-and-denoise: Protect large language model inference with local differential privacy," 2023, *arXiv:2310.09130*.

[142] F. Wu, Z. Li, Y. Li, B. Ding, and J. Gao, "FedBiOT: LLM local fine-tuning in federated learning without full model," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2024, pp. 3345–3355.

[143] W. Zhang et al., "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, Dec. 2021.

[144] I. Gim, C. Li, and L. Zhong, "Confidential prompting: Protecting user prompts from cloud LLM providers," 2024, *arXiv:2409.19134*.

[145] H. Watanabe and M. Uchikoshi, "Generating privacy-preserving personalized advice with zero-knowledge proofs and LLMs," in *Companion Proc. ACM Web Conf.*, May 2025, pp. 1385–1389.

[146] Z. Shi et al., "Privacy-enhancing paradigms within federated multi-agent systems," 2025, *arXiv:2503.08175*.

[147] X. Chen, L. Li, F. Ji, and W. Wu, "Memory-efficient split federated learning for LLM fine-tuning on heterogeneous mobile devices," 2025, *arXiv:2506.02940*.

[148] Z. Yu et al., "EDGE-LLM: Enabling efficient large language model adaptation on edge devices via unified compression and adaptive layer voting," in *Proc. 61st ACM/IEEE Design Autom. Conf.*, Jun. 2024, pp. 1–6.

[149] J. Zhang et al., "Towards building the federated GPT: Federated instruction tuning," 2023, *arXiv:2305.05644*.

[150] J. Yan, T. Chen, Y. Sun, Z. Nan, S. Zhou, and Z. Niu, "Mobility-aware asynchronous federated learning with dynamic sparsification," 2025, *arXiv:2506.07328*.

[151] S. Zhang, G. Cheng, W. Wu, X. Huang, L. Song, and X. Shen, "Split fine-tuning for large language models in wireless networks," *IEEE J. Sel. Topics Signal Process.*, early access, Jun. 19, 2025, doi: 10.1109/JSTSP.2025.3581484.

[152] Z. Lin et al., "SplitLoRA: A split parameter-efficient fine-tuning framework for large language models," 2024, *arXiv:2407.00952*.

[153] L. Che, J. Wang, Y. Zhou, and F. Ma, "Multimodal federated learning: A survey," *Sensors*, vol. 23, no. 15, p. 6986, Aug. 2023.

[154] L. Che, J. Wang, X. Liu, and F. Ma, "Leveraging foundation models for multi-modal federated learning with incomplete modality," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2024, pp. 401–417.

[155] O. Boyar, I. Priyadarsini, S. Takeda, and L. Hamada, "LLM-fusion: A novel multimodal fusion model for accelerated material discovery," 2025, *arXiv:2503.01022*.

[156] B. You, "Research on multimodal data fusion technology based on LLM and attention mechanism," in *Proc. 7th Int. Congr. Hum.-Comput. Interact., Optim. Robotic Appl. (ICHORA)*, May 2025, pp. 1–4.

[157] S. Li and H. Tang, "Multimodal alignment and fusion: A survey," 2024, *arXiv:2411.17040*.

[158] J. Bai et al., "Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond," 2023, *arXiv:2308.12966*.

[159] Q. Ma, M. Zhang, Y. Tang, and Z. Huang, "Att-sinkhorn: Multimodal alignment with sinkhorn-based deep attention architecture," in *Proc. 28th Int. Conf. Autom. Comput. (ICAC)*, Aug. 2023, pp. 1–6.

[160] M. F. M. Firdhous, W. Elbreiki, I. Abdullahi, B. H. Sudantha, and R. Budiarto, "WormGPT: A large language model chatbot for criminals," in *Proc. 24th Int. Arab Conf. Inf. Technol. (ACIT)*, Dec. 2023, pp. 1–6.

[161] H. Luo, G. Sun, H. Yu, B. Lei, and M. Guizani, "An energy-efficient wireless blockchain sharding scheme for PBFT consensus," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 3, pp. 3015–3027, May 2024.

[162] H. Luo, "ULS-PBFT: An ultra-low storage overhead PBFT consensus for blockchain," *Blockchain: Res. Appl.*, vol. 4, no. 4, Dec. 2023, Art. no. 100155.

[163] R. Chen, H. Luo, G. Sun, H. Yu, D. Niyato, and S. Dustdar, "DRDST: Low-latency DAG consensus through robust dynamic sharding and tree-broadcasting for IoV," 2024, *arXiv:2412.04742*.

[164] H. Luo, G. Sun, C. Chi, H. Yu, and M. Guizani, "Convergence of symbiotic communications and blockchain for sustainable and trustworthy 6G wireless networks," *IEEE Wireless Commun.*, vol. 32, no. 2, pp. 18–25, Apr. 2025.

[165] Y. Liu et al., "Blockchain-empowered lifecycle management for AI-generated content products in edge networks," *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 286–294, Jun. 2024.

[166] Y. Liu et al., "ProSecutor: Protecting mobile AIGC services on two-layer blockchain via reputation and contract theoretic approaches," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 10966–10983, Dec. 2024.

[167] Y. Lai, Y. Liu, H. Luo, G. Sun, C. Chi, and H. Yu, "Accelerating block and transaction propagation: A survey on broadcast protocols in blockchain networks," *IEEE Trans. Netw. Sci. Eng.*, early access, May 16, 2025, doi: 10.1109/TNSE.2025.3570870.

[168] T. Jiang et al., "Blockchain for energy market: A comprehensive survey," *Sustain. Energy, Grids Netw.*, vol. 41, Mar. 2025, Art. no. 101614.

[169] H. Luo et al., "Wireless blockchain meets 6G: The future trustworthy and ubiquitous connectivity," *IEEE Commun. Surveys & Tut.*, early access, Jul. 31, 2025, doi: 10.1109/COMST.2025.3593918.

[170] H. Luo, Q. Zhang, G. Sun, H. Yu, and D. Niyato, "Symbiotic blockchain consensus: Cognitive backscatter communications-enabled wireless blockchain consensus," *IEEE/ACM Trans. Netw.*, vol. 32, no. 6, pp. 5372–5387, Dec. 2024.

[171] B. Mao et al., "A blockchain-enabled cold start aggregation scheme for federated reinforcement learning-based task offloading in zero trust LEO satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 6, pp. 2172–2182, Jun. 2025.

[172] A. Singh et al., "FSPO: Few-shot preference optimization of synthetic preference data in LLMs elicits effective personalization to real users," 2025, *arXiv:2502.19312*.

[173] A. Radoi and G. Cioroiu, "Uncertainty-based learning of a lightweight model for multimodal emotion recognition," *IEEE Access*, vol. 12, pp. 120362–120374, 2024.

[174] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An enhanced MobileNet architecture," in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2019, pp. 0280–0285.

[175] L. Meegahapola, H. Hassoune, and D. Gatica-Perez, "M3BAT: Unsupervised domain adaptation for multimodal mobile sensing with multi-branch adversarial training," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 8, no. 2, pp. 1–30, May 2024.

[176] G. Sun et al., "Large language model (LLM)-enabled graphs in dynamic networking," *IEEE Netw.*, vol. 39, no. 4, pp. 290–301, Jul. 2025.

[177] Y. Zhao, Y. Xiu, C. Dai, N. Wei, and D. Niyato, "Movable antenna enhanced federated fine-tuning of large language models via hybrid client selection optimization," 2025, *arXiv:2506.00011*.

[178] T. Anthuvan and K. Maheshwari, "AI-C2C (conscious to conscience): A governance framework for ethical AI integration," *AI Ethics*, vol. 5, no. 4, pp. 1–13, Aug. 2025.

[179] Y. Liu et al., "Secure multi-LLM agentic AI and agentification for edge general intelligence by zero-trust: A survey," 2025, *arXiv:2508.19870*.

[180] R. Zhang et al., "Covert prompt transmission for secure large language model services," 2025, *arXiv:2504.21311*.

[181] W. Wang et al., "Retrieval-augmented perception: High-resolution image perception meets visual RAG," 2025, *arXiv:2503.01222*.

[182] R. Zhang et al., "Interactive AI with retrieval-augmented generation for next generation networking," *IEEE Netw.*, vol. 38, no. 6, pp. 414–424, Nov. 2024.

[183] X. Wang et al., "Chain-of-thought for large language model-empowered wireless communications," 2025, *arXiv:2505.22320*.

[184] C. Zhao et al., "World models for cognitive agents: Transforming edge intelligence in future networks," 2025, *arXiv:2506.00417*.

[185] G. Liu et al., "Wireless agentic AI with retrieval-augmented multimodal semantic perception," 2025, *arXiv:2505.23275*.

[186] R. Zhang et al., "Optimizing generative AI networking: A dual perspective with multi-agent systems and mixture of experts," 2024, *arXiv:2405.12472*.