Received 31 May, 2023; revised 6 September, 2023; accepted 24 September, 2023.

Digital Object Identifier 10.1109/OJCOMS.2023.3320646

An Overview on Generative AI at Scale with Edge-Cloud Computing

YUN-CHENG WANG (STUDENT MEMBER, IEEE), JINTANG XUE (STUDENT MEMBER, IEEE), CHENGWEI WEI (STUDENT MEMBER, IEEE), AND C.-C. JAY KUO (FELLOW, IEEE)

[1]University of Southern California, Los Angeles, California, USA

CORRESPONDING AUTHOR: YUN-CHENG WANG (e-mail: yunchenw@usc.edu).

ABSTRACT As a specific category of artificial intelligence (AI), generative artificial intelligence (GenAI) generates new content that resembles what humans create. The rapid development of GenAI systems has created a huge amount of new data on the Internet, posing new challenges to current computing and communication frameworks. Currently, GenAI services rely on the traditional cloud computing framework due to the need for large computation resources. However, such services will encounter high latency because of data transmission and a high volume of user requests. On the other hand, edge-cloud computing can provide adequate computation power and low latency at the same time through the collaboration between edges and the cloud. Thus, it is attractive to build GenAI systems at scale by leveraging the edge-cloud computing paradigm. In this overview paper, we review recent developments in GenAI and edge-cloud computing, respectively. Then, we use two exemplary GenAI applications to discuss technical challenges in scaling up their solutions using edge-cloud collaborative systems. Finally, we list design considerations for training and deploying GenAI systems at scale and point out future research directions.

INDEX TERMS Artificial intelligence, AI-generated content, edge-cloud computing, distributed system, lightweight models, Metaverse, artificial intelligence of things.

I. INTRODUCTION

GENERATIVE AI (GenAI) has emerged as a groundbreaking field to realize artificial general intelligence (AGI) by integrating machine learning and creative content generation. It is a specific category of AI that aims to autonomously generate new content that imitates the content created by humans in different modalities, including images [42, 94], audio [101, 100], text [32, 13], and even 3D objects [81, 87]. With the rapid development of GenAI, various applications, such as text-to-image generation [68, 116], text-to-speech (TTS) synthesis [65, 144], chatbot [1, 79], and AI-empowered mixed reality (MR) [98, 137], have been widely used by consumers. Recently, GenAI models rely on deep neural networks, such as generative adversarial networks (GANs) [39] and large language models (LLMs) [10] because of the higher complexity of the generative tasks. As a result, such GenAI models have huge model sizes and are computationally demanding, a powerful centralized computation infrastructure (i.e., cloud server) is required to process requests from users. Thus, users may experience high latency if the cloud experiences a high volume of traffic. Such limitations hinder the applicability of GenAI to applications with low latency requirements. Besides, the heavy computation in a cloud consumes a significant amount of energy. The overly centralized computing framework is eco-unfriendly, unsustainable, and cost-inefficient.

In recent years, the proliferation of mobile devices and the exponential growth of data-intensive applications have spurred the development of edge-cloud computing solutions. Edge-cloud computing takes advantage of powerful computation resources in cloud servers and efficient data management and communication in edge servers. It has emerged as a promising solution for consumer-based AI applications and edge intelligence. For example, several large AI models are deployed with the edge-cloud computing system [93, 134]. Compared to traditional cloud computing and multi-access edge computing (MEC), edge-cloud computing can exploit more computation resources and achieve lower latency through the collaboration between clouds and edges.

GenAI poses unprecedented challenges to scalable computing systems and the need for edge-cloud computing because of three main reasons: 1) a significant amount of data generated, 2) consumer-centric applications, and 3) high cost to maintain centralized GenAI services. First, compared

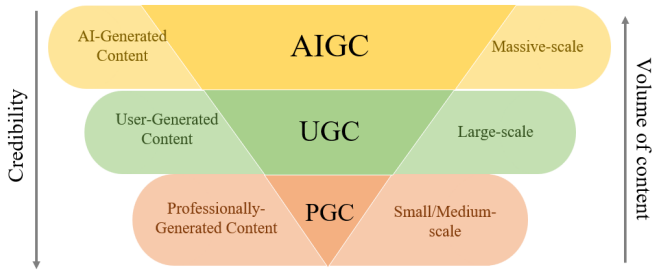


FIGURE 1: The significant amount of data generated in the AIGC era poses an unprecedented challenge in computer networks.

to discriminative AI, GenAI produces a significant amount of multimedia content, or so-called AI-generated content (AIGC), in different modalities, such as audio, images, text, etc. Fig. 1 shows the evolution of different phases in content creation. Compared to professionally-generated content (PGC) and user-generated content (UGC), GenAI created much more data on the Internet. As a result, transmission latency becomes a serious challenge in GenAI services. Although latency is a common challenge of deploying models at the edge, it is even more so in the context of GenAI due to a much larger data amount.

The second challenge is the unique application domain of GenAI. Currently, most GenAI services target consumer-centric applications. In addition, many applications require real-time interactions, such as the chatbots. It makes more sense to place the computation system closer to users instead of relying on a centralized computation infrastructure to process all user requests. In addition, edge-cloud computing can preserve more privacy for users by storing their data only on local servers or user devices. Deploying GenAI services closer to the users by adopting an edge-cloud computing paradigm can improve efficiency and data privacy.

Third, the required resources to run GenAI services are huge. For example, ChatGPT by OpenAI[1] is one of the most popular GenAI services recently. It is a chatbot used to interactively answer users' questions in human-like responses. It processed more than 13 million daily requests in January 2023 [133]. Although the exact computing infrastructure used by the ChatGPT service is not publicly available, we can estimate the cost to run the service each day based on the model architecture of GPT-3 [10], the generative model to support the ChatGPT service. GPT-3 is an LLM containing 175 billion parameters, which requires more than 350 GB of RAM and VRAM to run the model. To deploy such a large model with minimum latency, a distributed parallel computing system with at least 2,048 GPUs is required [133, 10] to handle user inputs. Relying solely on the computation power in the cloud would lead to high latency when the request volume is high. In addition, its daily electricity charge is estimated to be around $600,000 using NVIDIA A100 GPUs; not to mention the training of GPT-3, which requires 10^8 times computation and more than 10^5 iterations. It is neither cost-efficient nor feasible to deploy such a service entirely on the cloud servers.

Due to the above-mentioned three emerging challenges, the collaboration of edge and cloud computing resources will mitigate the burden of cloud servers, especially under the high volume of requests, or "at scale". In this paper, we examine four important aspects of deploying GenAI under edge-cloud computing: 1) computation and data offloading, 2) low latency, 3) personalization, and 4) privacy. Our main contributions are summarized below:

- Provision of a comprehensive overview of recent developments in both GenAI models and edge-cloud computing;
- Identification of technical challenges in training and deploying large-scale GenAI services using today's solution;
- Presentation of design considerations for training and deploying GenAI that target computational efficiency (i.e., lower power consumption), low latency, personalization, and privacy;
- Visualization of two large-scale GenAI applications as concrete examples to support our discussion;
- Future research directions on GenAI systems based on edge-cloud computing.

The rest of this paper is organized as below. A comparison of this work and related previous overview papers is made in Sec. II. Sec. III introduces the background of GenAI and defines the scope for "GenAI at scale". Reviews on recent developments of GenAI models and edge-cloud computing are conducted in Sec. IV. Two application scenarios are envisioned in Sec. V. Technical challenges in training and deploying GenAI systems at scale with current distributed systems are examined in Sec. VI. Design considerations to address them with edge-cloud computing are elaborated in Sec. VII. Finally, future directions are pointed out in Sec. VIII, and concluding remarks are given in Sec. IX.

II. COMPARISON WITH RELATED WORK

We summarize related overview papers and compare them with this work in Table. 1. This work is the first one devoted to GenAI services in the edge-cloud computing paradigm, and it includes network design considerations and guidance for future research.

A. OVERVIEW ON GENERATIVE AI

After the release of ChatGPT at the end of 2022, interest in GenAI increased rapidly, and a number of survey or overview papers on GenAI have been published [12, 133, 142]. Some focus on how GenAI models can be applied to different applications, such as audio diffusion [144], text-to-image generation [143], and multimodality [115] applications. Nev-

[1] https://openai.com/blog/chatgpt

TABLE 1: Comparison between our work and other related survey and overview papers.

Year	Reference	Contributions	GenAI	Edge Intelligence	System Design
2020	[110]	Introduce communication-efficient techniques from both algorithmic and system perspectives.	x	v	v
2021	[84]	Introduce communication-efficient techniques from both algorithmic and system perspectives.	x	v	x
2022	[139]	Summarize major research efforts where machine learning systems have been deployed at the edge of computer networks.	x	v	x
2023	[142]	Review fundamental GenAI techniques and applications in different modalities.	v	x	x
2023	[12]	Survey on the basic components of GenAI, recent advances, and applications of uni-modality and multi-modality GenAI models.	v	x	x
2023	[136]	Deployment of AIGC network and mobile applications via collaborative edge-cloud infrastructure.	v	v	x
2023	Ours	Review on both GenAI models and edge intelligence; point out challenges and bottlenecks in current GenAI services; propose design considerations to address the issues; provide future directions on how edge-cloud computing can benefit GenAI..	v	v	v

ertheless, most of them are concerned with the algorithmic aspect of GenAI. Here, we study technical challenges related to the deployment of the entire GenAI systems at scale and propose a practical cloud-edge computing solution.

B. OVERVIEW ON EDGE INTELLIGENCE

There are plenty of survey and overview papers on AI in edge-cloud computing, or so-called "edge intelligence". Most of them consider discriminative AI tasks, where the systems only need to make binary decisions. For example, one important topic for security is how to apply edge intelligence in surveillance cameras [139, 50, 84, 110]. Other emerging edge intelligence applications include unmanned autonomous vehicles (UAV) [82] and the Internet of Things (IoT) [34]. The latter has long been an important field since the 5G and wireless networks arrived. A roadmap about the integration of edge-cloud computing and AI is given in [23].

GenAI has become a new application domain of AI technology in recent years. It poses emerging challenges, including a huge amount of machine-created content, large model sizes and power consumption, and low latency requirements in real-time applications, such as GenAI for gaming. It is a critical problem since the amount of transmitted content is much more than discriminant AI tasks. To the best of our knowledge, [136] is the only work that addressed GenAI at the edge. However, it focused on the review of existing papers. In this work, we not only provide a comprehensive review of recent developments of GenAI and edge-cloud computing but also have an in-depth discussion on many related issues, including technical challenges, design considerations, exemplary applications, and future technology outlook of GenAI deployment at scale using the edge-cloud platform.

III. BACKGROUND

A. BRIEF HISTORY OF GENERATIVE AI

Before discussing the necessity of deploying scalable GenAI services at the edge today, we present a brief history of GenAI, which can be roughly divided into the following four stages:

- 1950 $\sim$ 1990: Expert systems;
- 1990 $\sim$ 2020: Deep neural networks;
- 2020 $\sim$ 2023: Proprietary cloud computing;
- 2023 $\sim$ - : Public edge-cloud computing.

When the concept of AI was introduced in the earliest stage (1950 $\sim$ 1990), people were fascinated by the GenAI idea since it could model human-like interactions. Compared to discriminative AI, where only low-dimensional decision vectors were predicted, the GenAI technology was not mature enough to offer powerful GenAI services at that time. Most human-like interactions were configured in the form of rule-based expert systems [52] and/or template fillings [3].

Through persistent efforts over the three decades in the second stage (1990s $\sim$ 2020s), deep neural networks became more powerful and popular. Researchers applied them to GenAI and had several breakthroughs [39]. However, since GenAI was still in the development and prototyping stage by the research community, its scalability was not a concern.

Recently, several commercial companies have started to develop their own GenAI services using large language/image/video models and proprietary data. The performance of such services is impressive due to the adopted large model sizes and a huge amount of training data. The services are typically deployed on cloud computing systems with powerful computation resources. Proprietary GenAI systems have concerns in various aspects such as privacy, power consumption, and model efficiency. First, since most services

are closed-sourced and proprietary, user privacy protection cannot be well enforced. The model development process and the final developed models are not transparent. Second, the power consumption of cloud servers for running deep neural networks and transformers [123] is high. Third, since all computations are conducted in centralized computing facilities, long physical distances between data sources and end users tend to yield high latency. Real-time applications are difficult to achieve.

GenAI has entered the commercial usage stage with large exposure to the general public. Due to the proliferation of mobile and edge devices, the data sources and computation should be placed as close to the user as possible to reduce communication latency and improve user privacy. We envision the next stage of GenAI should be open-sourced services that adopt an edge-cloud computing paradigm. To accommodate an increasing number of daily users, scalability and sustainability are serious technical and business issues in deploying future GenAI services. In this paper, we target such accessible, affordable, and sustainable GenAI services, providing feasible solutions to domain-specific GenAI applications.

B. DEFINITION OF GENERATIVE AI AT SCALE

As the load to certain services increases, the services should maintain a constant response time in the face of this increased workload because new nodes are added to the cluster, and new server instances are run. Such a data service is called a scalable one. When the services fail to meet the requirements in a centralized cloud cluster, the edge-cloud computing paradigm can provide significant benefits in time, computation, and power efficiency. To understand how GenAI services work at scale in the edge-cloud paradigm, we must examine the available computation power, network speed, number of concurrent connections or users, and latency requirements, as discussed below.

a) Memory. A modern cloud computing infrastructure often contains thousands of GPUs as computing power. Each GPU's video RAM (VRAM) can range from 32 to 80 GB. Thus, the total number of memory available is at the scale of 100TB. An LLM, LLaMA, containing 7B $\sim$ 65B model parameters require 28GB $\sim$ 260GB of VRAM to process an inference request. In other words, the cloud server can only handle $\sim$ 4,000 requests at once, even using the most lightweight model. However, usually, for a cloud server, there will be as many as 500,000 concurrent requests, which are much more than what it can process in real-time. Then, any GenAI models that require more than 200MB of memory during inference demand distributed processing at scale.

b) Network Bandwidth & Concurrent Connections. One unique characteristic of GenAI services is the output dimensions. Their output dimensions are much larger than discriminative AI services since the latter only outputs low-dimensional decision vectors. For example, the output dimension can be up to 1920$\times$1080, equivalent to 6MB of data in image generation tasks, while the output dimension is between 96kbps $\sim$ 160kbps in audio synthesis tasks. The transmitted AI-generated content can easily exceed the network bandwidth of a centralized cluster. Any GenAI models that transmit generated content exceed the network bandwidth, where

$$\textit{output bitrate} \times \textit{\# of connections} \geq \textit{bandwidth},$$

demand distributed processing at scale.

c) Computation & Latency. For real-time applications, such as dialogue agents and Metaverse with AI-generated scenes, latency is one of the top priorities. Latency is also closely related to the computation power and the number of FLOPs. For example, a 90fps frame rate is required to avoid dizziness in the Metaverse, meaning that the computation resources should be powerful enough to generate the content in 1/90 second. Considering using A100 GPUs, the computation power is 312 teraFLOPs per second. To meet the 90 fps requirement, the model needs to have a number of FLOPs lower than 3.5 teraFLOPs to achieve real-time interactions.

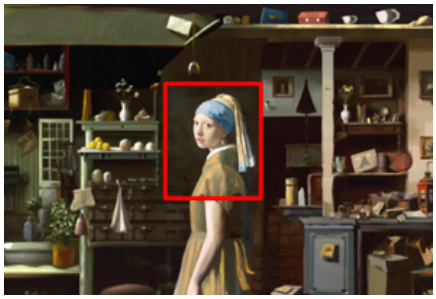
These requirements should be jointly considered, and one will affect the other. For example, the number of concurrent connections will affect the required network bandwidth and latency; the number of model parameters to fit in the computation infrastructure will affect the number of concurrent connections.

IV. RECENT DEVELOPMENTS IN GENERATIVE AI AND EDGE-CLOUD COMPUTING

A. GENERATIVE AI

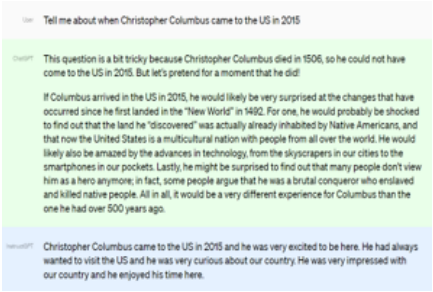
With the explosion of ChatGPT, GenAI has become a hot topic. GenAI is an AI technology that can generate various multimedia content [88, 46, 115]. Fig. 2 shows some real-world applications of GenAI, including images, texts, audio, graphics, and even 3D objects. The historical development of generative AI can be roughly divided into three eras: 1) the Variational Autoencoder (VAE) and Generative Adversarial Network (GAN) era (2014-2017), 2) the Transformer era (2018-2019), and 3) the large model era (2020-present) [35]. Three popular architectures for GenAI models are shown in Fig. 3.

a) Variational Autoencoder (VAE). The Variational Autoencoder (VAE) was first proposed in [62]. It has several variations [131, 38, 141, 4] to improve the quality of the generated content [120, 67], adjust to different levels of supervision [36], and improve the inference efficiency [112]. VAEs are probabilistic generative models. Their encoder and decoder correspond to two neural networks. The encoder maps an input to a vector in a latent space, while the decoder maps a latent vector back to the input space to generate an output. In the training stage, the network parameters are optimized so that the output is as close as possible to the input. Adding noise to latent vectors makes the decoder produce multiple output samples that have the same distribution as input samples.



(a) Image outpainting[3].

(b) Image captioning.



(c) Chatbot[1].

(d) Text-to-speech (TTS)

(e) AI art.

(f) Metaverse.

FIGURE 2: Six illustrative applications of GenAI models: a) image outpainting, b) image captioning, c) chatbot, d) text-to-speech, e) AI art, and f) Metaverse.

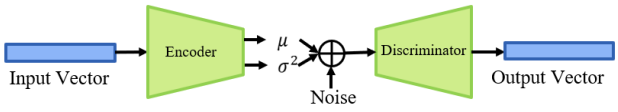
b) Generative Adversarial Network (GAN). Similar to VAE, Generative Adversarial Networks (GANs) [39] need two networks in the training stage but keep only one in the inference stage [129, 89, 49, 21]. The two networks are a generator and a discriminator. Through a training process [44, 53], the generator generates better and better fake data that are getting closer to real data in the distribution to fool the discriminator. On the other hand, the discriminator is used to differentiate real and fake data as much as possible. The generator and discriminator are trained by solving a min-max optimization problem:

$$\min_G \max_D V(G,D) = \mathop{\mathbb{E}}_{x\sim\text{real}}[\log D(x)] + \mathop{\mathbb{E}}_{G(z)\sim\text{fake}}[1-\log D(G(z))],$$

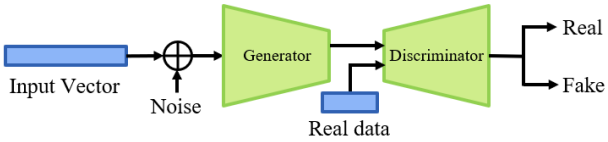
where $G(*)$ and $D(*)$ denotes the generator and discriminator, respectively. Its capability improves along the training process. Gradually, they reach an equilibrium status where fake and real data are so close that they cannot be easily differentiated. Then, the training stage is completed.

c) Transformers. Natural language generation (NLG) models aim to generate human-like textual responses. There are several common applications, such as neural machine translation [17, 114], question answering [20, 138], and document summarization [124, 85]. Such models are also called language models (LMs) [130]. In recent years, transformers [123] with self-attention mechanisms have made major breakthroughs in establishing powerful LMs [45, 59, 118, 78, 107]. Transformers have replaced the long short-term memory (LSTM) [48] as the preferred LM architecture and set off a new wave of large language models (LLMs) [75, 56, 145, 27]. They often adopt an encoder-decoder architecture, as shown in Fig. 3 (c). While the encoder adopts a bi-directional information propagation process to understand the input text, the decoder in most transformer architectures generates words one by one. Such a decoder is also called the autoregressive decoder. With the advent of transformers, generative models are getting larger and larger. Over the past two years, attempts have been made to combine a wide variety of models to create larger and more powerful models. They offer impressive performance in various fields [41]. Due to the large model sizes of GenAI models, they are deployed on the cloud nowadays. That is, models are trained at the training stage and run at the inference stage in cloud servers. Users send requests to the cloud server for content generation. Then, the generated content is sent back to users.

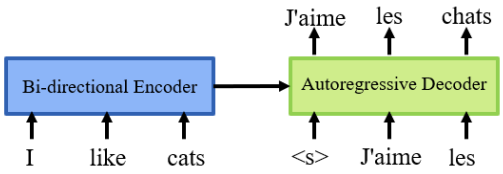
d) Online Services & Scalability. However, due to the long distance between users and the cloud, the above-mentioned framework is not scalable. It has a higher generation latency, which hinders specific applications such as augmented reality (AR) / virtual reality (VR) / mixed reality (MR). Furthermore, with the rapid growth of GenAI services, the amount of AI-generated data on the Internet has increased significantly (see Fig. 1). Some GenAI web-based



(a) Variational Auto-encoder (VAE).



(b) Generative Adversarial Network (GAN).



(c) Transformers.

FIGURE 3: Architectures of three popular GenAI model categories: VAE, GAN, and Transformers.

services, such as mid-journey[2] and DALL-E[3], have a large number of users per day. Most GenAI services are not free since the required computation is costly due to high power consumption. Latency can be another major concern once the service becomes popular with growing user requests.

It is worthwhile to emphasize that user feedback is important for model fine-tuning. In other words, there are interactions between the cloud and the edges. Besides, collaboration among users is important for training a more robust and diverse system. Edge-cloud computing provides a natural solution to build GenAI systems at scale. Yet, to the best of our knowledge, there is no research addressing how a distributed system should be designed to accommodate the computation, transmission, and exchange of a huge amount of AIGC data. This motivates us to explore this topic and write this overview paper.

B. EDGE-CLOUD COMPUTING

There are three basic paradigms for implementing large-scale computing systems. They are: 1) cloud computing, 2) multi-access edge computing (MEC), or previously mobile-edge computing, and 3) edge-cloud computing, as shown in Fig. 4. Among the three, cloud computing carries out computationally demanding projects using a large number of online servers to serve many remote users. The cloud has much larger computing resources than a local site. Moving compute-intensive tasks to the cloud has been an efficient way of data processing. The concept of cloud computing was introduced in the early 60s [37, 113]. It has made rapid progress in the last several decades and has become a mature business service model. Examples include: Amazon Web Services (AWS)[4], Microsoft Azure[5], Google Cloud Platform (GCP)[6], IBM Cloud[7], Salesforce[8], etc.

As the computational power of mobile devices increases and wireless networks become accessible at almost any place, multi-access edge computing (MEC) provides computing, storage, and bandwidth closer to users. MEC tends to allocate more computing tasks to the edge than the cloud. Computation can be performed near data sources on edge devices. Edge computing has become more important nowadays, as pointed out in a few studies, e.g., [28, 80, 109, 11, 29]. The MEC framework primarily relies on edge devices, which have limited resources. In addition, the MEC framework greatly relies on caching to improve the latency. Thus, its performance is not good for computationally demanding tasks.

As the demand for real-time processing, low-latency communication, and efficient data management increases, the edge-cloud computing paradigm emerges as a new and attractive solution. By combining the power of cloud computing with the proximity and responsiveness of edge devices, edge-cloud computing aims to bridge the gap between latency and scalability. Since it has lower latency, it is suitable for real-time applications such as AR/VR/MR [147, 31], object tracking and detection [99, 122], etc. Since it can utilize computational resources at both the cloud and edges, it has more flexibility in load balancing to yield a more scalable solution. Moreover, user data and privacy can be better preserved by edge-cloud computing [91].

The hardware and performance specifications of three computational resources (namely, cloud servers, edge servers, and user devices) are compared in Table 2. As shown in the table, cloud servers have the highest resources in

TABLE 2: Comparison of hardware and performance specifications of three computational resources, namely cloud servers, edge servers, and user devices.

Resources	Cloud Servers	Edge Servers	User Devices
Memory	>24TB	∼500GB	<64GB
Dist Storage	>25PB	<1PB	<10TB
Latency (RTTs)	30 ∼ 50 ms	<10ms	-
Power (per year)	>2,000TWh	∼7,500KWh	∼600KWh
Concurrent Connections	>500,000	∼1,000	1

[2]https://www.midjourney.com/

[3]https://openai.com/blog/dall-e-introducing-outpainting

[4]http://aws.amazon.com/ec2

[5]http://www.microsoft.com/azure

[6]https://cloud.google.com/

[7]https://www.ibm.com/cloud

[8]https://www.salesforce.com/

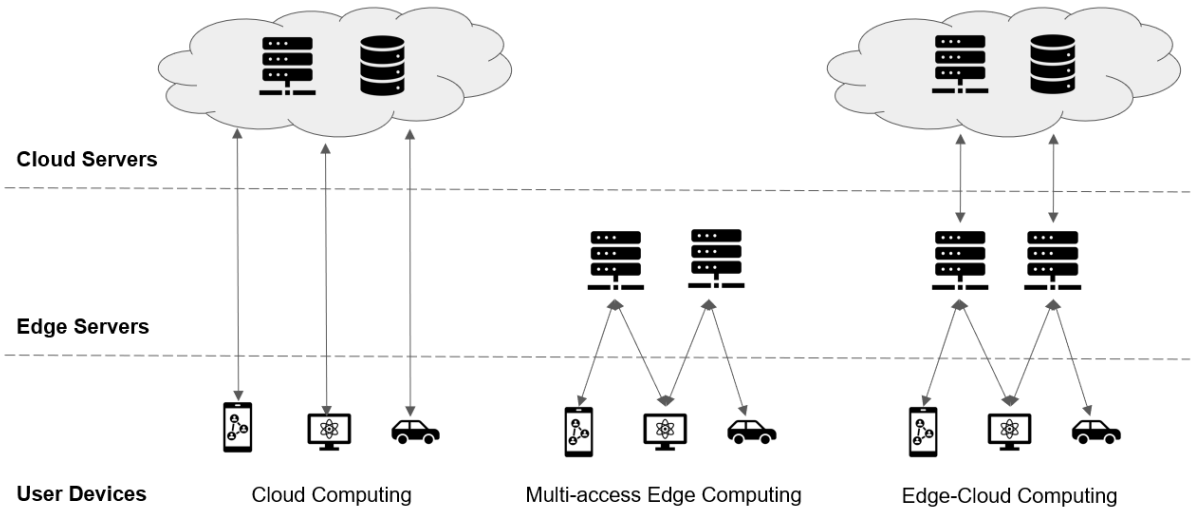


FIGURE 4: Three basic computing paradigms in support of large-scale computing systems.

terms of computational memory and data storage capacity. At the same time, they have the highest power consumption and the largest number of concurrent connections. Their latency is also the highest since they are far from users. It is beneficial to shift some computation loads from cloud servers to edge servers and user devices to balance the computational load and reduce latency in various applications. The load-balancing idea is also called offloading. Computation offloading [33, 51] and data offloading [150, 47] are two key concepts in edge-cloud computing.

The AI tasks suitable for edge servers and cloud servers are shown in Fig. 5. Due to rich computation resources, cloud servers can store and run large models to process high-level tasks. In contrast, edge devices are mainly responsible for low-level pre-processing tasks. Due to the emergence of 5G/IoT, AIGC enters a new era. That is, it is no longer sufficient to conduct all computations and store all data in a centralized cloud server or data center. Similarly, AI computation with edge servers and user devices is also not practical in building a scalable system as AIGC data grows fast.

Some large deep-learning AI models are difficult to deploy at the edges. Recently, a green learning methodology [66] has been proposed as an alternative to deep learning. Green learning AI models have much smaller model sizes, significantly lower computational complexity in terms of FLOPs (Floating Point Operations), faster inference time, and less power consumption demand. As a result, green-learning AI models open a new door for edge servers and even user devices in offloading cloud servers. Hybrid deep- and green-learning solutions match the edge-cloud computing paradigm well. That is, GenAI has a unique mission to process low-level data and aggregate high-level abstractions to generate creative content. GenAI can benefit the most from the collaboration of edge and cloud servers.

Recently, Meta announced a supercomputing cluster with very rich computational resources[9]. It can perform five exaflops (billion billion calculations per second) using a total of 16,000 NVIDIA A100 GPUs to train state-of-the-art GenAI models. Servers are connected by an NVIDIA Quantum InfiniBand fabric network with a bandwidth of 16 Tb/s to ensure low latency in data synchronization. However, this computational scale is not affordable for most companies and academic institutions. Thus, how to design scalable GenAI systems using a reasonable computing cluster to perform similar tasks is of great interest. We put hope in edge-cloud computing since it can leverage expandable computation resources that are under-utilized and closer to users.

The deployment of GenAI systems on the edge-cloud computing platform is shown in Fig. 6. Since the training of GenAI models is most computationally heavy, it is still conducted in cloud servers. The training is usually done offline and asynchronous. The deployment of trained GenAI models for the AIGC tasks can be placed as close to users as possible to lower latency. Edge servers can be used to fine-tune GenAI models, train personalized models, preserve user privacy, and serve as an interface between edges and cloud servers. It is ideal to have several edge servers to handle individual tasks separately. More details about the design considerations will be discussed in Sec. VII.

V. TWO EXEMPLARY SERVICES

In this section, we present two exemplary GenAI services as concrete examples to demonstrate how to deploy GenAI in the edge-cloud computing environment. In particular, these services demand low latency and will have a large number of

[9]https://ai.facebook.com/blog/ai-rsc/

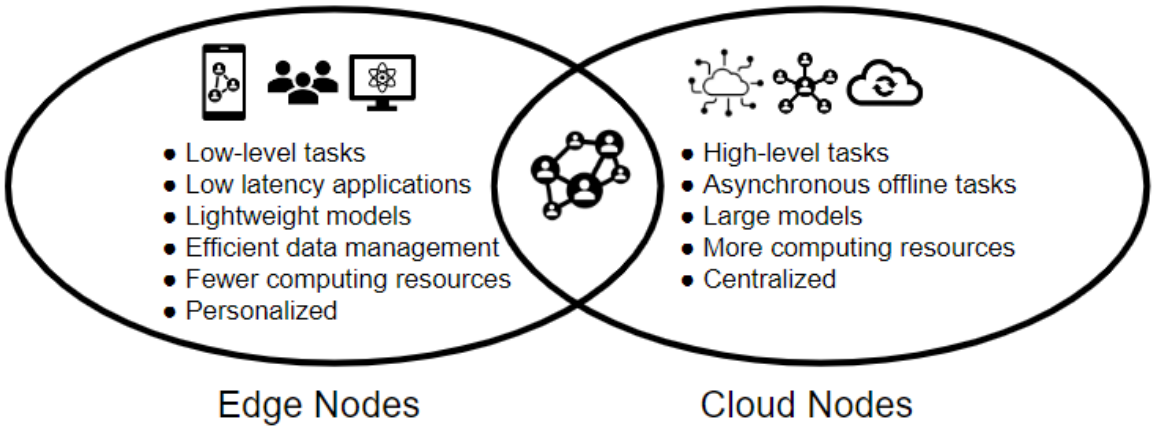


FIGURE 5: Roles and suitable applications for edge nodes and cloud nodes in edge-cloud computing.

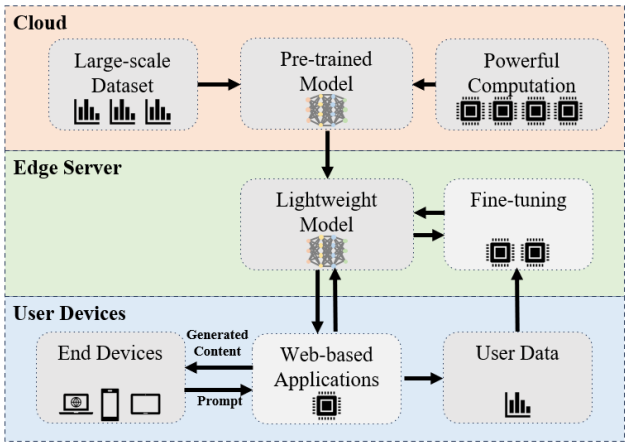


FIGURE 6: Implementation of GenAI systems with the edge-cloud computing paradigm.

users when the technologies become mature and the markets are ready. Scalability-based edge-cloud computing is critical to their successful deployment. They are, a) Metaverse system, which is a performance- and latency-centric application, and b) artificial intelligence of things (AIoT), which is a personalization- and privacy-centric application. Details of GenAI model deployment in the cloud and edges are given separately below.

A. METAVERSE SYSTEM

Metaverse is one of the most important applications in GenAI. With the development of GenAI, most of the generated scenes rely on machine learning models. Metaverse requires an extremely low latency to make the transition smoother in order to avoid dizziness. However, high-quality rendering is time-consuming, and virtual reality (VR) goggles are resource-constrained. Generating satisfactory scenes and meeting the low latency requirement with resource-constrained edge devices is the key to the success of the Metaverse system. Apparently, its solutions at scale demand the close collaboration of the computation resources at the edges and the cloud.

In the Metaverse system, every user should be placed in a single virtual environment. As a result, a huge map will be required to be generated. Edge-cloud computing can be a latency-efficient solution for Metaverse applications. For example, as illustrated in Fig. 7, the entire map is stored in the centralized data center that can be shared among all users. Then, the locations, angles, and other parameters can be collected by user devices and transmitted through a wireless network [55]. The cloud computing clusters are also responsible for generating the scenes and rendering the results. The compressed scenes will be sent back to the users. At the user end, a lightweight decoder and renderer are deployed to display the scenes based on the corresponding viewpoints of the users. As a result, such a system design can reduce the latency significantly since the computation-heavy parts are taken care of using powerful computation infrastructure. In addition, the amount of data transmitted in the communication systems is minimized. The users will send the request to the GenAI models in the cloud, and the compressed scenes will be transmitted back to the users.

Edge servers are a fundamental component in the Metaverse system [73]. They serve a similar role as in the content delivery network (CDN) to distribute content based on geographical locations and share the computation load in the cloud server. Users in the same locations will be connected to the same edge server. Once a user sends a request to the Metaverse system to generate the local scene, it is transmitted through the edge servers and cached. Other users in the same location can access the cached scenes in the edge servers to further reduce the latency. Computation resources in the edge servers should also be leveraged. For example, they can be helpful in compressing and decompressing the scenes

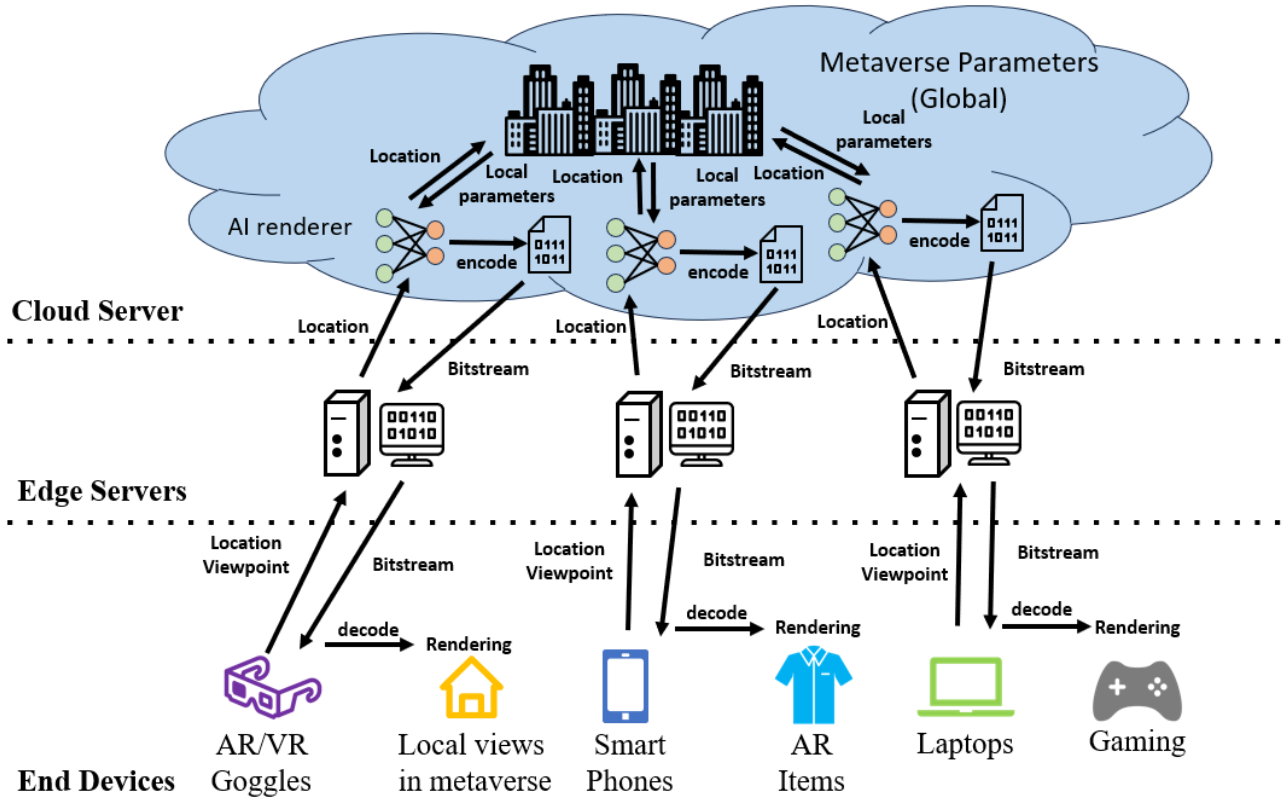


FIGURE 7: Illustration of exemplary service of the Metaverse system with GenAI under edge-cloud computing.

generated in the cloud server. As a result, not only the latency can be reduced, but also the quality of the generated scenes is improved.

B. ARTIFICIAL INTELLIGENCE OF THINGS

Artificial Intelligence of Things (AIoT) is an emerging application to combine artificial intelligence (AI) technologies in the Internet of Things (IoT) systems [146]. Through the integration of AI and ubiquitous wireless networking infrastructure, one can build AIoT systems where the end devices have certain intelligence in data processing and analytics. GenAI can be further exploited to facilitate a broader range of applications. For example, a voiced assistant can interact with users in applications such as autonomous driving, smart cities, and smart homes, where fluent human speech has to be automatically generated from multiple information sources, which is often in the form of text data.

To implement AIoT with edge-cloud computing (say, voiced assistant applications), we need to consider privacy, personalization, and data synchronization [16]. Users may collect data to train more relevant personalized GenAI models. Training a simple GenAI model with acceptable performance on user devices is desired. Then, model parameters of multiple users can be sent to cloud servers to be integrated to build a more advanced GenAI model through federated learning [86, 58] or split learning [106, 149]. In addition, data can be constantly collected from the end devices to ensure the information in GenAI models is up-to-date. Online optimization [72, 132] supports GenAI model training with streams of data on the fly. User devices can be synchronized with advanced GenAI models through firmware updates. As a result, the whole system can benefit from a larger pool of training data from users via federated or split learning while user data privacy can be well protected.

The hierarchy in edge-cloud computing can be utilized for more efficient GenAI model deployment. For example, large, middle-size, and lightweight models can be placed in cloud servers, edge serves, and user devices, respectively. Different resolutions of the models can be achieved through knowledge distillation [126, 40] and model parameter pruning [104, 55]. Grouping users with the same computation facility can further reduce the computation. Different edge and cloud servers can be specialized to process different applications efficiently. Personalization can be considered to optimize end devices according to user behavior. The personalization fine-tuning on the user devices is generally efficient due to the deployment of lightweight models.

VI. TECHNICAL CHALLENGES

There are technical challenges in training and deploying GenAI services at scale. The major ones include: 1) increased output dimensions, 2) growth in model sizes, 3)

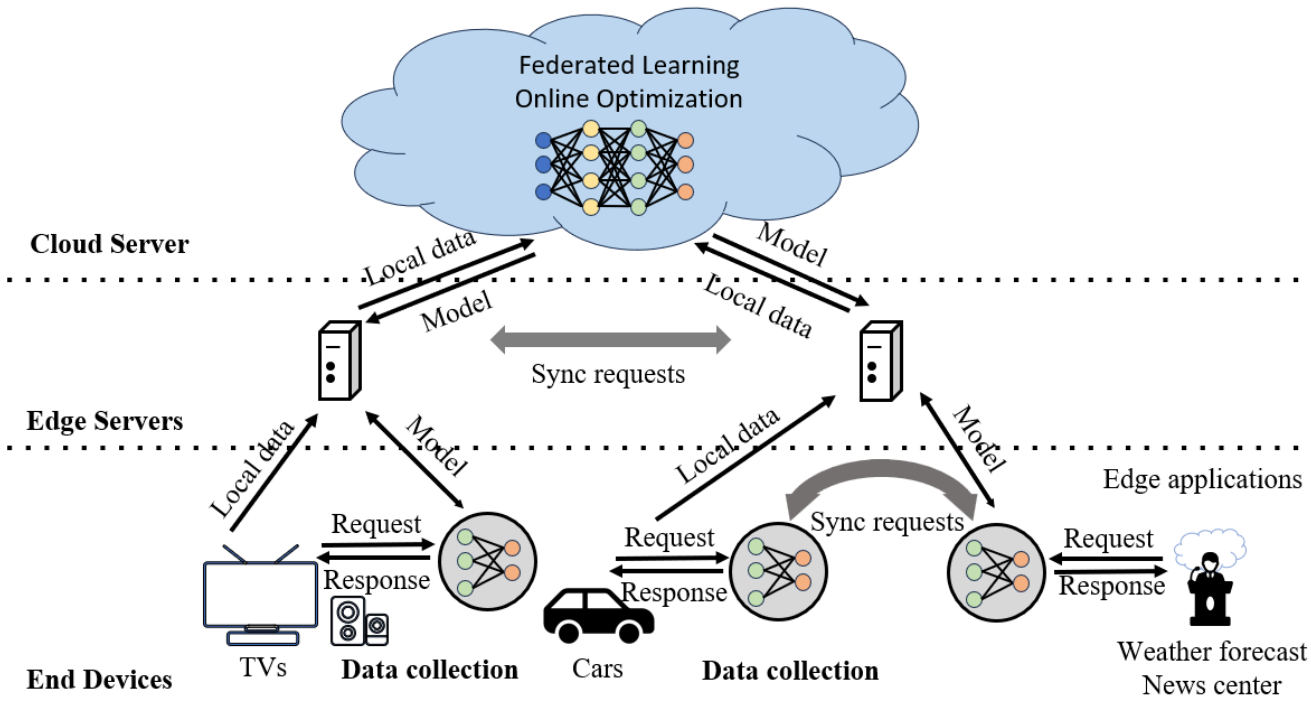


FIGURE 8: Illustration of exemplary service of the AIoT system with GenAI in the edge-cloud computing environment.

power consumption, 4) latency, and 5) infrastructure reliability. They are summarized below to demonstrate the need for good resource coordination between edges and the cloud with edge-cloud computing.

A. INCREASED OUTPUT DIMENSIONS

GenAI is a specific category of AI that creates new content in multimedia formats, such as audio, images, or texts. Compared to discriminative AI, the output dimensions of GenAI are much larger, posing a new challenge in transmitting a high volume of data. For example, for discriminative AI, the outputs are usually a low-dimensional vector, say, the decision vector. They can be easily transmitted even with a large number of requests. In contrast, it is challenging to transmit a high volume of multimedia data from the cloud center to users for GenAI services. Data compression techniques [127] are needed in GenAI. In addition, GenAI usually has a larger model size than discriminative AI. The former often demands Transformers [123], while the latter may adopt convolutional neural networks. Consequently, GenAI demands more computational resources, including hardware costs and power consumption. Thus, designing an efficient edge-cloud GenAI system at scale is a unique challenge.

B. GROWTH IN MODEL SIZES

In order to achieve better performance in various applications, GenAI systems adopt larger models with more model parameters and computation over time. The growth rate of

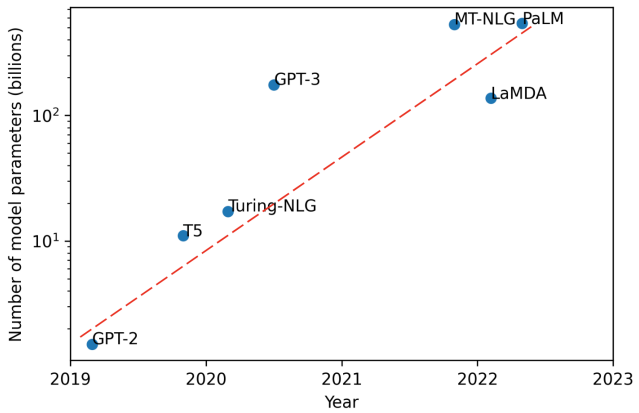
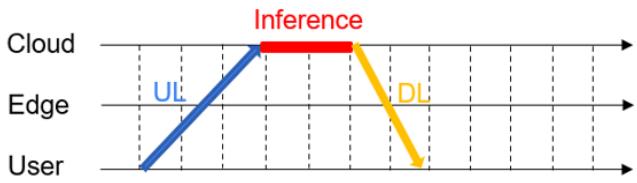
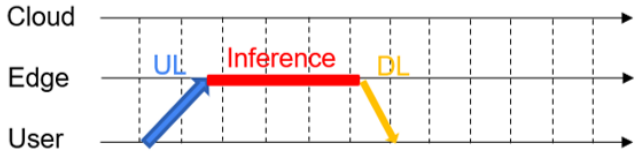


FIGURE 9: The development of generative LLMs and their model sizes as a function of time. The vertical axis is in log scale. Models in the figure include GPT-2 [96], T5 [97], Turing-NLG [111], GPT-3 [10], LaMDA [119], MT-NLG [111], and PaLM [18].

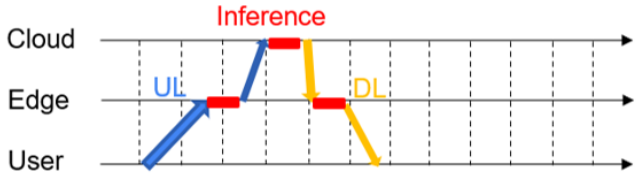
their model sizes is an exponential function of time [57] as shown in Fig. 9. Specifically, the model sizes of neural GenAI models double every 6 months as reported in [12]. This is called "Moore's Law for GenAI". In contrast, the computation power of CPUs and GPUs only doubles every two years in the semiconductor manufacturing industry. If the trend continues, the demand for computation will surpass its supply in the near future. Unless there is a



(a) Cloud Computing.



(b) Multi-access Edge Computing (MEC).



(c) Edge-cloud Computing.

FIGURE 10: Illustration of latency in different computation frameworks.

major breakthrough in supply, its limitation will hinder the future growth of GenAI systems. Thus, how to train and run GenAI systems through collaboration between the cloud and edges efficiently has become an urgent issue for the entire community to tackle.

C. POWER CONSUMPTION

Power consumption is a major concern in cloud computing [135, 83]. The centralized computation infrastructure consumes a significant amount of electricity in running user requests as well as training large models. Fig. 3 compares power consumption, carbon emission, and cloud computational cost in training large GenAI models for different modalities. The power consumption of GenAI services is even greater than simply training GenAI models since they need to process millions of requests per day from the users.

Power consumption and carbon emission are closely related to the number of floating point operations (FLOPs). More FLOPs imply higher carbon emissions and electricity bills. For example, the GPT-3 model, the backbone of ChatGPT, demands 10^{23} FLOPs in one training iteration and 10^{15} FLOPs in inference. Since the power efficiency of CPUs/GPUs in modern computation facilities is around 10^{10} FLOPs/sec-watt, it will demand 27.78 kWh (10^5 Joule) to process a single request. Apparently, GenAI services are not scalable. Furthermore, they are eco-unfriendly, unsustainable, and cost-inefficient. To achieve sustainability with large-scale GenAI services, alternative Green solutions under the edge-cloud computing paradigm are essential.

D. LATENCY

For real-time GenAI applications such as VR and gaming, it is of uttermost importance to reduce latency. The latency calculation in three different computing frameworks is illustrated in Fig. 10. It is the time between a request sent and its response received at the user end. It is determined by uplink transmission time, inference time, and downlink transmission time; namely,

$$latency = t_{UL} + t_{inference} + t_{DL},$$

where t_{UL}, $t_{inference}$, and t_{DL} denote uplink transmission time, inference time, and downlink transmission time, respectively. In the cloud computing framework, the latency comes from the long uplink transmission time t_{UL} and downlink transmission time t_{DL}. since the computation resources are placed far from the users. In MEC, the transmission delay is reduced since the processing units are placed closer to the users. However, the computation resources in edge servers are not as powerful as the ones in the cloud servers. Thus, the inference time, $t_{inference}$ will be much longer, especially for computation-intensive applications, such as GenAI services. In edge-cloud computing, tasks are divided efficiently between the edge and cloud servers. Thus, the overall inference delay can be reduced by leveraging both computation resources in edge and cloud servers. In addition, the transmission delay is also reduced since the connection between edge servers and the cloud is much faster than from the user end.

For GenAI applications, their inference time can be longer than that of other applications due to larger model sizes and more computations required by GenAI models. Furthermore, the output of GenAI services can be multimedia AIGC. Transmission of multimedia data such as video will demand a longer downlink transmission time t_{DL} than text data. We can reduce t_{DL} by allocating multimedia generation tasks to edge servers. Again, the development of green-learning-based GenAI models are in urgent need.

E. INFRASTRUCTURE RELIABILITY

Cloud servers need a large number of GPUs to handle user requests at scale. As mentioned before, Meta has just started a supercomputing center with 16,000 NVIDIA A100 GPUs to support their GenAI services. It is unrealistic to set up such powerful but costly infrastructures in many sites globally. Furthermore, such a huge single-site infrastructure is vulnerable to physical and/or cyberspace attacks. Distributed computing with multiple lightweight cloud servers and much more edge servers will offer a more robust AI computational infrastructure in the future.

TABLE 3: Comparison of power consumption, carbon emission, and cloud computational cost in the training of large GenAI models in different modalities.

Model	Modality	Hardware	Power (watts)	Hours	Energy Comsumption (kWh)	CO_2e (lbs)
WaveGAN [24]	Audio	P100 GPU x1	250	96	24	19.63
GANSynth [30]	Audio	V100 GPU x1	300	108	32.4	26.5
FloWaveNet [60]	Audio	V100 GPU x1	300	272	81.6	66.74
BigGAN [9]	Image	V100 GPU x1	300	3,072	921.3	753.54
Stable Diffusion [102]	Image	V100 GPU x1	300	2,184	655	535.72
GPT-2 [96]	Text	TPUv3 x 32	-	168	2.8×10^4	2.39×10^4
GPT-3 [10]	Text	V100 GPU x10,000	-	355	1.29×10^6	1.1×10^6
GLaM [92]	Text	TPUv4s	-	-	4.56×10^5	8×10^4

VII. SYSTEM DESIGN CONSIDERATIONS

Design considerations for providing GenAI services at scale using edge-cloud computing are examined in this section. Training and deployment of GenAI services should be considered separately. For the training of GenAI models, a larger amount of computational resources and training data are needed. Key considerations include: 1) computation offloading, 2) personalization, 3) privacy, and 4) information recency. After models are trained, it is desired to deploy them on user devices for lower latency and power consumption. There are three main considerations: 1) lightweight models, 2) minimizing latency through edge-cloud collaboration, and 3) multi-modality content generation and interface. First, lightweight models are essential because of limited resources on edge servers and user devices. Second, by properly dividing the inference tasks to edges and the cloud, inference latency can be largely reduced through edge-cloud collaboration. Third, multimedia content will become the main media for humans to acquire information, as evidenced by the popularity of videos on the Internet nowadays. Multi-modality content generation and interface at edges should be considered carefully. Fig. 11 summarizes the design considerations for providing GenAI services at scale.

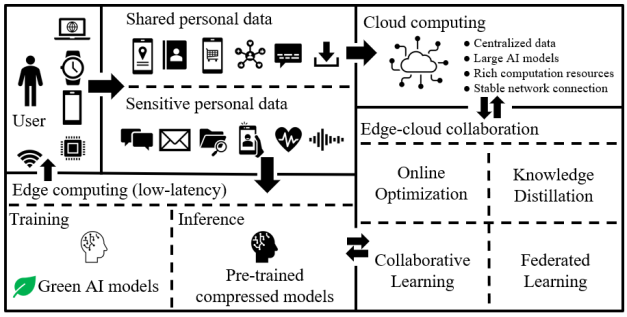


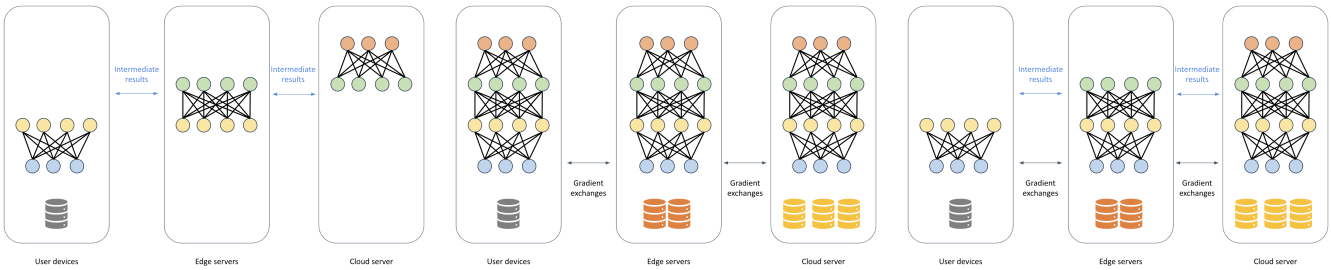
FIGURE 11: The roadmap of designing GenAI services at scale. Computation offloading, latency, privacy, and data offloading are the major considerations.

A. TRAINING

Since the training of large-scale GenAI models is costly, we need to consider the following issues.

a) Computation offloading. This is an important concept in edge-cloud computing and collaboration. It means that we need to fully utilize computation resources in the cloud and edges. Traditional cloud computing puts all computational loads in a centralized cluster. Users might experience long latency if the resources in the cloud cannot meet the requirements of sudden heavy service requests. Furthermore, the computational cost to train large GenAI models is extremely high. It may take days or weeks to train large models. Thus, computation offloading has to be considered when training GenAI systems under the edge-cloud computing paradigm.

Most GenAI services adopt deep neural networks (DNNs) as models. DNNs consist of multiple layers. To balance computation loads in training DNNs, we can decouple the training procedure. [77] illustrates how DNNs can be trained by different workers in parallelism, as shown in Fig. 12. Such an idea can be leveraged in edge-cloud computing, where the user devices, edge servers, and the cloud serve as different workers. Thus, in edge-cloud computing, data does not need to be entirely transmitted to the cloud server, and the training does not need to take place entirely in the cloud. Instead, different layers can be trained by different computational facilities (e.g. user devices, edge servers, and the cloud server). For example, as shown in Fig. 12 (a), deeper layers are farthest from users, and they can be trained in the cloud. Gradients are propagated to edge servers to train middle layers. Gradients are propagated again to user devices. Finally, shallow layers are closest to users, and their parameters can be trained on user devices. As a result, system optimization can be carried out through the collaboration of user devices, edge servers, and the cloud server. Only the gradient information has to be transmitted in such a design. Another idea is to decouple the training data as shown in Fig. 12 (b). Smaller DNNs can be trained in parallel by leveraging data parallelism. Then, multiple smaller models can be integrated through federated learning. Finally, a hybrid solution exploiting both model parallelism and data parallelism can be explored as well as shown in Fig. 12 (c). Under the GenAI context, such parallelism



(a) Model parallelism. (b) Data parallelism. (c) Hybrid parallelism.

FIGURE 12: Three parallelism for computation and data offloading in DNN model training [77].

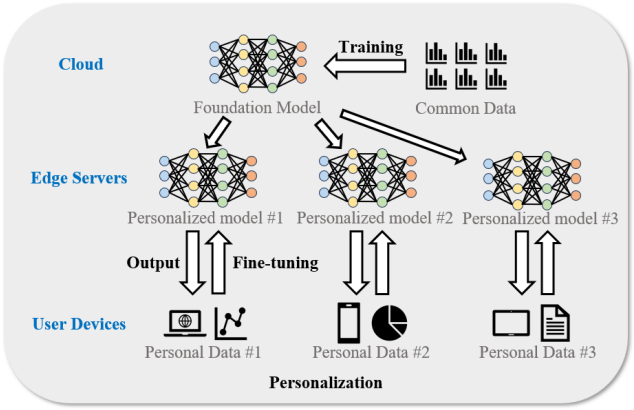


FIGURE 13: Personalization of GenAI services.

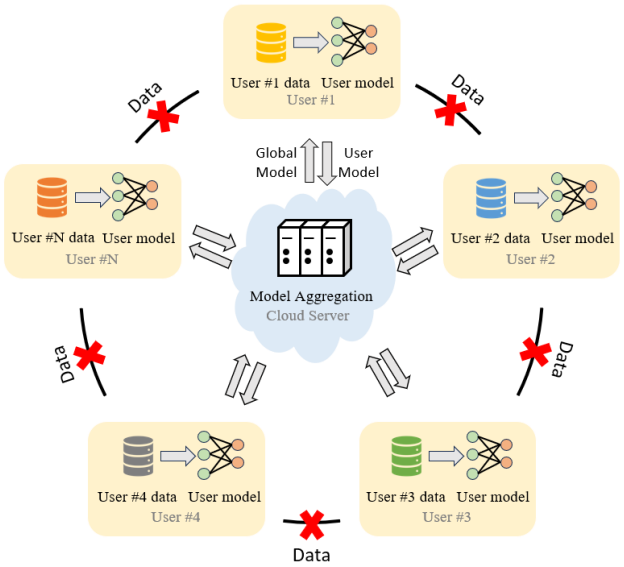


FIGURE 14: Privacy preservation through federated learning.

and collaboration between edges and the cloud are even more important. Computation and data offloading should be carefully designed in large-scale GenAI services.

b) Personalization. Edge-cloud computing can provide personalized GenAI models. While training a GenAI model requires a large amount of data, personalization can be achieved by fine-tuning the trained model with a small amount of user data. The collaboration between edges and the cloud for personalized services is depicted in Fig. 13. First, an advanced GenAI model, called the foundation model, should be trained in the cloud with common data. In this step, the trained foundation model can handle general requests. To achieve personalization, personal data, such as user logs and metadata, are collected from user devices and sent to edge servers. The foundation model is also placed in edge servers for personalization. Then, a fine-tuning technique can be developed to shift the model domain from a generic one to a user-specific one using personal data. Typically, fine-tuning requires much fewer computation resources, and it can be entirely conducted in edge servers.

c) Privacy. Privacy is a major concern in GenAI services to prevent personal information from being disclosed to other users and companies. It is particularly important in the context of GenAI services since generated content is difficult to control. One solution to privacy is the use of federated learning, as shown in Fig. 14. The core concept is to share the model parameters among users instead of sharing personal data. Users will have their own models stored in user devices or edge servers based on applications. The models are trained based on user data. Information exchange among users is through aggregating user models in the cloud. That is, all trained user models are transmitted from edges to the cloud, where small user models are combined to train an advanced large model. Finally, the model parameters of the advanced model will be synchronized with user models for the next round of training. By sharing model parameters in federated learning, GenAI services can preserve user privacy while collecting relevant information from users.

Besides federated learning, split learning [5, 106, 149] offers a powerful solution to data privacy preservation when training GenAI models in a distributed setting. Instead of passing model parameters as done in federated learning, split

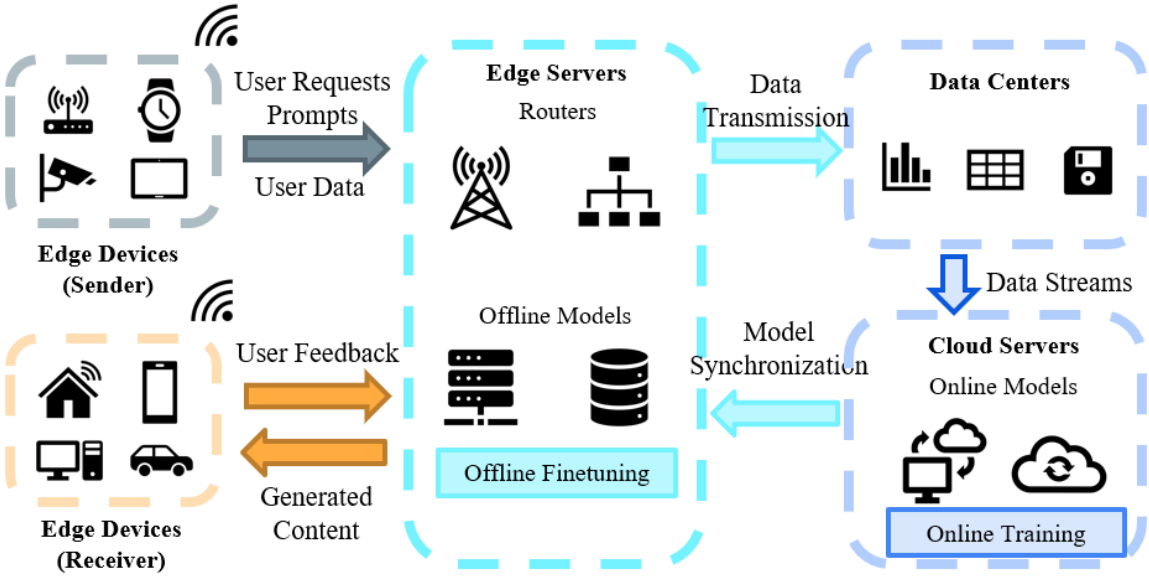


FIGURE 15: Online optimization in edge-cloud computing.

learning shares the gradients among different sections of the models that are trained by different clients independently. Thus, no other clients can access the original raw data. In such a way, models can be optimized with the arrival of new data samples, while data privacy is preserved at the same time, in an edge-cloud collaborative fashion.

d) Information Recency. Keeping the information updated is one of the main challenges to GenAI services. For example, chatbots need the most updated information to offer a better user experience. On the other hand, training GenAI models is time-consuming and inefficient. Incremental learning is needed. However, it is not easy to implement in neural network models. Online optimization with edge-cloud computing is an alternative way to keep the services updated. This is illustrated in Fig. 15. Usually, it contains two models - an online model and an offline model. The online model is stored in the cloud server for the most updated information by adopting online optimization. At the same time, a smaller offline model is placed in the edge servers for low latency inference and cloud online model offloading. Online and offline models are synchronized periodically to ensure that edge intelligence is also up-to-date.

B. DEPLOYMENT

Three design considerations in deploying GenAI services are elaborated below.

a) Lightweight Models. Deploying GenAI models on edge servers and user devices can lower latency in user-centric applications. Large GenAI models cannot be deployed on user devices due to their large model sizes and high power consumption. Lightweight GenAI models, as summarized in Fig. 16 are more suitable. For example, knowledge distillation can fit into edge-cloud computing well. With knowledge distillation, the knowledge learned in a huge teacher model is transferred to a smaller student model. Thus, the teacher model can be trained and stored in the cloud server while the student model is distilled from the teacher model in the edge servers and, then, stored in user devices. Model pruning adopts a similar concept to train a smaller model from a large model, which takes place in edge servers. Other techniques include quantization and model compression. They can reduce the model sizes effectively without the collaboration between the cloud and edges.

Recently, there has been an increasing number of research focusing on developing lightweight GenAI models. LLaMA [121] reduces the number of model parameters in LLMs to as small as 7 billion using a self-instruct training technique called Alpaca [117]. Lightweight GenAI models encourage the development of mobile- or web-based applications on user devices, such as WebLLM[10]. The small model sizes also alleviate the burden in caching-based communication networks. Latency is also largely reduced due to lower computation and transmission delay. Research in developing lightweight GenAI models demonstrates the urgency to reduce the ridiculously large models while still having comparable performance.

b) Minimizing latency through edge-cloud collaboration. When deploying GenAI models, it is desirable to minimize the latency through the collaboration between edge and cloud servers as illustrated in Fig. 10. First, transmission latency is largely reduced due to the introduction of edge servers [14]. In general, applications are sped up 20 times while reducing energy consumption by 5% [19]. Further-

[10]https://mlc.ai/web-llm/

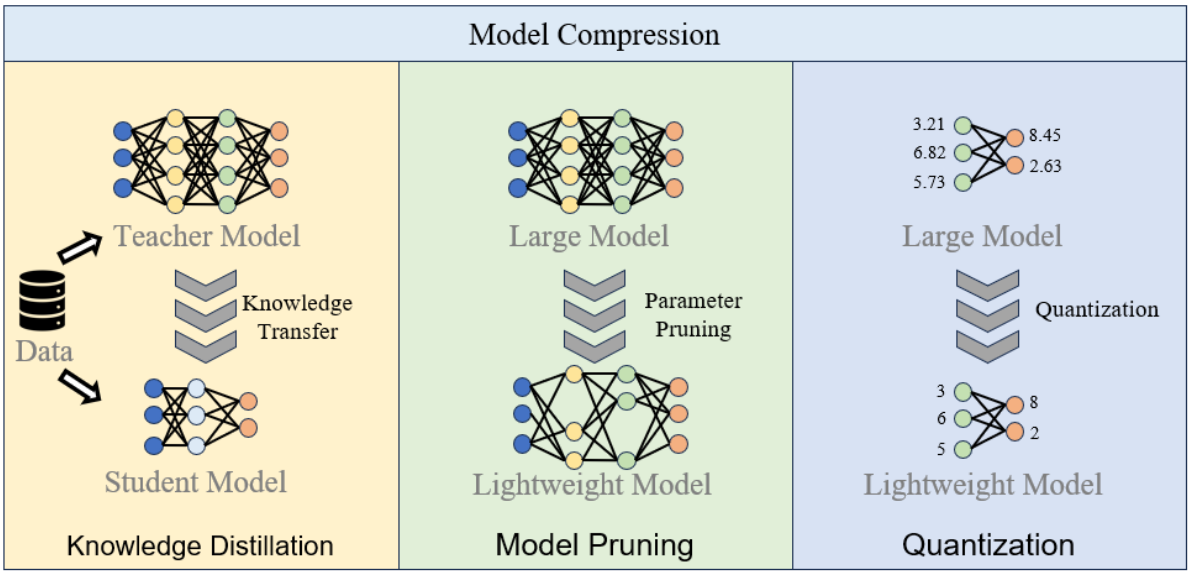


FIGURE 16: Existing technologies to obtain lightweight GenAI models.

more, an optimized strategy to divide the computation tasks among edges and the cloud can reduce the inference latency. This is critical to scalable and efficient model deployment.

We attempt to analyze the inference latency of GenAI models in the edge-cloud computing system by proposing clear instructions on how the tasks should be divided among edges and the cloud below. To estimate the inference latency, edge and cloud servers can be modeled as M/M/c queues since most servers adopt parallel computing using GPUs. Each inference request can be modeled as a customer in the queue. There are two important parameters to specify for each inference job:

- **FLOPs (F)** governs the service rate of the server. A higher FLOPs indicate a longer service time under the same computation resources.
- **Memory Usage (U)** governs the number of parallel jobs to be run at the server. A higher memory usage will lead to fewer concurrent jobs.

At the server end, there are three important parameters:

- **GPU Memory (G)** controls how many jobs can be run simultaneously.
- **Computation Power (P)** controls the service rate.
- **Concurrent Connections (N)** controls the arrival rate.

We can specify an M/M/c queue as:

$$c = G \setminus U, \quad \lambda = N, \quad \mu = P \setminus F,$$

where λ is the arrival rate and μ is the service rate. For example, LLaMA is a powerful task generation model. The smallest model contains 7B parameters. It will consume about 28GB of memory during inference. Inference of LLaMA-7B will require around 13.1 GFLOPs. Suppose the cloud server is equipped with 100 NVIDIA A100 GPUs, and the edge server is equipped with 8 NVIDIA V100 GPUs. Each A100 GPU has 80GB of memory, and it can process 312T FLOPs per second. Each V100 GPU has 32GB memory, and it can process 120T FLOPs per second. The cloud server has an arrival rate of 1,000, while the edge server has an arrival rate of 20. Then, the cloud server can be modeled as an M/M/285 queue with $\lambda = 1000$ and $\mu = 23816$, and the edge server can be modeled as an M/M/9 queue with $\lambda = 20$ and $\mu = 9160$. As a result, the cloud server has a higher server utilization to be able to handle multiple concurrent jobs efficiently. On the other hand, the edge servers have a lower arrival rate and work load so it is efficient to process jobs sequentially.

Furthermore, the average service time W in the M/M/c queue can be written as

$$W = \frac{1}{\mu} + \frac{C(c, \frac{\lambda}{\mu})}{c\mu - \lambda},$$

where

$$C(c, \frac{\lambda}{\mu}) = \frac{1}{1 + (1 - \frac{\lambda}{c\mu}) \frac{\mu^c c!}{\lambda^c} \Sigma_{k=0}^{c-1} \frac{\lambda^k}{\mu^k k!}}$$

is referred to as Erlang's C formula. The inference latency of an edge server is bottlenecked by its service rate $\frac{1}{\mu}$. Consequently, to minimize the overall inference latency, tasks with lower FLOPs but higher memory usage, such as preprocessing tasks, should be distributed to edges, and tasks with higher FLOPs but lower memory usage, such as deep neural networks, should be distributed to the cloud.

c) Multi-modality Content Generation and Interface. Image captioning and text-to-image generation are two examples of multi-modality content generation and interface. To implement multi-modality content generation, we need a joint embedding space to connect two different modalities.

CLIP [95] is a well-known multi-modality GenAI model. It learns a joint multi-modal latent space for language and vision through contrastive pre-training. We elaborate on how such a framework can be efficiently deployed under the edge-cloud computing paradigm. The multi-modality models usually consist of three modules: 1) the input module, 2) the generation model, and 3) the output module. The first and third modules are more relevant to users, and they do not require as many computational resources as the second module. Thus, we can place the input/output modules in edge servers or user devices to avoid transmitting generated content. The main generation module is deployed in the cloud server since it requires more computation resources.

VIII. FUTURE RESEARCH DIRECTIONS

It is important to think beyond the current GenAI service framework in proposing future research directions. Some promising topics are given in this section.

A. GENERIC VERSUS DOMAIN-SPECIFIC GENERATIVE AI MODELS

As one of the most famous GenAI services nowadays, ChatGPT provides a generic GenAI model at the expense of a large model size and a high running cost. It may be advantageous to trade breadth for depth of generated content to lower the service cost and enhance the quality of services. That is, instead of handling general questions, it is more efficient to train GenAI models in a specific domain. In addition to parameter efficiency in training domain-specific models, domain-specific applications imply more homogeneous data and users. As a result, under the edge-cloud computing paradigm, it is more likely to adopt caching [64] to further improve efficiency. Examples of domain-specific applications include healthcare, financial advice, etc., where the accuracy of generated content is the top priority in some application domains.

B. DECOMPOSITION OF LARGE LANGUAGE MODELS

ChatGPT is a large language model (LLM) built upon large pre-trained transformers for generative tasks. It does not leverage the tool of knowledge graphs (KGs), where knowledge is stored in a graph-structured format. It is appealing to decompose a large language model into smaller ones that have an interface with domain-specific KGs. This decomposition is expected to lower the complexity of the GenAI system for cost reduction. The resulting AIGC services can be more transparent and scalable. Furthermore, personalization is easier to offer with the help of KGs [105]. That is, generic KGs are stored in the cloud, while personalized KGs are stored in local servers or user devices. In addition, edge and cloud servers can collaborate in a way that the reasoning tasks using LLMs are processed in the cloud with more computational resources, while the edge servers are responsible for natural language understanding (NLU) and natural language generation (NLG) with constrained resources.

C. QUALITY AIGC ASSURANCE

The quality assessment of the generated content, i.e., how similar are they to the human-generated content, is an important future research topic. Such quality assurance modules can be easily deployed at user devices as the filter of the content generated in the cloud. The quality assurance module can be trained collaboratively with the GenAI models in the cloud to improve the performance capability [74]. We may have different considerations against different AIGC modalities. Two examples are given below.

a) Visual Content. One may use common sense to evaluate the quality of generated visual content. For example, a picture with a person riding a horse is more natural than the opposite. Generated content that contradicts common sense tends to look strange to users. Sensitive content, copyright content, and trademarks should also be avoided in the generated content [148, 25, 26]. Automatic detection [43, 63] of strange and/or forbidden AIGC is still an open problem. Furthermore, deepfake images can be a security concern for some applications. A lightweight deep fake detection solution [15] has been developed to address this concern.

b) Textual Content. The quality of generated texts can be evaluated at three levels: grammatical correctness, readability, and factual correctness. Coherency and conciseness are criteria for readability. They are more difficult to evaluate than grammatical errors. Mis/disinformation is already common over the Internet. It will be even easier to generate a large amount of fake news for malicious purposes with the GenAI service.

D. GREEN GENERATIVE AI MODELS

To address the high carbon footprint yielded by huge deep learning networks, green learning [66] has been proposed as an alternative learning paradigm in recent years. A green learning model is characterized by its low carbon footprint, lightweight model, low computational complexity, and logical transparency. In addition, unlike deep neural networks, which require end-to-end optimization, green learning models are modularized and can be optimized separately. Such a characteristic is particularly appealing under edge-cloud collaboration as individual modules can be optimized at the user devices with minimum memory requirement and carbon footprint. Green GenAI models have been explored in the last several years, e.g., NITES [70], TGHop [71], Pager [6], GENHOP [69]. These models are very attractive at the edges. They can also be implemented in cloud servers to reduce carbon footprints and save electricity bills. More efforts along this line are needed.

E. ATTACKS AND DEFENSE

Attacks and defenses are important in computer networks and AI models. From the communication perspective, since most of the computations are conducted in the cloud servers, user data will be transmitted from user devices to edges, then finally to clouds. In addition, the generated content will be sent back to the users. In such a process, the data will travel through many computers and networks, increasing the risk of backdoor attacks on the models or the data. It is important to design a defense mechanism [8] for the generated content, such as data encryption [76], to prevent any attack during transmission. From the model perspective, the generated content can be manipulated to yield harmful outcomes [108, 22]. Such an attack on the GenAI models is called an adversarial attack. Thus, detecting adversarial attacks and improving the robustness and trustworthiness of GenAI models are essential.

F. HIERARCHICAL KNOWLEDGE SYSTEM

"*Does GenAI have the intelligence to understand user requests?*" There has been a heated debate for a while about how GenAI models understand user inputs and react to them. However, like humans, there is no intelligent agents can be built without a knowledge system. In the world of computers, knowledge systems are usually represented as knowledge graphs (KGs) [54], which store knowledge in a graph format. To achieve artificial general intelligence (AGI), a mechanism for the models to communicate with KGs is required and demands further investigation [140, 125]. KGs are usually stored as databases and can interact with GenAI models efficiently. In addition, the agent in the cloud and the agent on the user devices may not need the same degree of the knowledge system due to the different hardware specifications. Cloud agents can serve as "teacher models" and are equipped with more universal knowledge, while edge agents usually only need to focus on a specific and customized task so the knowledge system can be efficiently distilled and learn from the teacher models [105, 128]. Such a hierarchical knowledge system is important to achieve AGI, especially under the edge-cloud computing paradigm.

G. COLLABORATION AMONG DIFFERENT AGENCIES

It is a unique characteristic of edge-cloud computing to achieve user and data privacy and data collaboration in model training at the same time. Such a characteristic is especially crucial for several specific application domains that cherish these two requirements. For example, one practical application domain is for the public sector [2]. While under different bureaus, the data are confidential and cannot be shared among each other. However, it is very often that collaboration between different bureaus is required for training a better GenAI system in the public sector. Other examples are GenAI for education [7] and GenAI for hospitals [61]. While the data among different institutions should not be shared with each other, common knowledge can be exchanged through an edge-cloud collaboration paradigm. These are the practical examples for the future application domains for GenAI under edge-cloud computing.

H. BIAS AND FAIRNESS

Bias and fairness have been important topics in AI research [103, 90] for a long time. They are even more important for GenAI since the generated multimedia content might be affected by the bias more easily than discriminative AI. The bias factors include cultural differences, differences in application domains, etc. They may come from differences in large training corpora collected and stored in the cloud and in distributed training data collected by user devices from different population groups. For example, the chatbot might be trained primarily on English data in the cloud, and, as a result, it has a bias against low-resource languages with poorer performance. Healthcare-oriented GenAI is particularly concerned with issues of bias and fairness since the corresponding professional services have high liability and demand high accuracy. Through edge cloud collaboration [139], it is possible to mitigate the bias and fairness issue in GenAI since the information is shared among cloud and edge servers, which allows a broader range of data sources.

IX. CONCLUSION

The training and deployment of GenAI services at scale pose a new challenge to the design of modern edge-cloud computational systems due to extremely large model sizes, increased output dimensions, heavy power consumption, and potential latency caused by a lack of computational and network resources. Two illustrative GenAI services were envisioned to show the importance of developing GenAI systems at scale on the one hand and validate the challenging claims on the other hand in this work. Afterward, an in-depth discussion on various design considerations of GenAI services over current communication systems were given. It was concluded that a desired design has to balance computational resources between edges and cloud servers and consider latency, data privacy, and personalization. Specifically, federated and split learning, where small GenAI models are trained at edges, while large GenAI models are trained at the cloud by combining a large number of small models, are expected to play important roles. As a result, most inference tasks can be distributed at edges. Finally, we point out several future research directions, such as domain-specific GenAI models, decomposition of large language models, green GenAI models, quality AIGC assurance, attacks and defense in edge-cloud computing, hierarchical knowledge system, collaboration between different, and bias and fairness of GenAI.

REFERENCES

[1] E. Adamopoulou and L. Moussiades, "An overview of chatbot technology," in *IFIP international conference on artificial intelligence applications and innovations*. Springer, 2020, pp. 373–383.

[2] N. Aoki, "An experimental study of public trust in ai chatbots in the public sector," *Government Information Quarterly*, vol. 37, no. 4, p. 101490, 2020.

[3] S. Arora, K. Batra, and S. Singh, "Dialogue system: A brief review," *arXiv preprint arXiv:1306.4134*, 2013.

[4] A. Asperti, D. Evangelista, and E. Loli Piccolomini, "A survey on variational autoencoders from a green ai perspective," *SN Computer Science*, vol. 2, no. 4, p. 301, 2021.

[5] A. Ayad, M. Renner, and A. Schmeink, "Improving the communication and computation efficiency of split learning for iot applications," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 01–06.

[6] Z. Azizi and C.-C. J. Kuo, "Pager: Progressive attribute-guided extendable robust image generation," *arXiv preprint arXiv:2206.00162*, 2022.

[7] D. Baidoo-Anu and L. Owusu Ansah, "Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of chatgpt in promoting teaching and learning," *Available at SSRN 4337484*, 2023.

[8] C. Barrett, B. Boyd, E. Burzstein, N. Carlini, B. Chen, J. Choi, A. R. Chowdhury, M. Christodorescu, A. Datta, S. Feizi *et al.*, "Identifying and mitigating the security risks of generative ai," *arXiv preprint arXiv:2308.14840*, 2023.

[9] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[11] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE access*, vol. 8, pp. 85 714–85 728, 2020.

[12] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt," *arXiv preprint arXiv:2303.04226*, 2023.

[13] A. Celikyilmaz, E. Clark, and J. Gao, "Evaluation of text generation: A survey," *arXiv preprint arXiv:2006.14799*, 2020.

[14] B. Charyyev, E. Arslan, and M. H. Gunes, "Latency comparison of cloud datacenters and edge servers," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[15] H.-S. Chen, M. Rouhsedaghat, H. Ghani, S. Hu, S. You, and C.-C. J. Kuo, "Defakehop: A light-weight high-performance deepfake detector," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[16] P. Cheng and U. Roedig, "Personal voice assistant security and privacy—a survey," *Proceedings of the IEEE*, vol. 110, no. 4, pp. 476–507, 2022.

[17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation. arxiv 2014," *arXiv preprint arXiv:1406.1078*, 2020.

[18] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[19] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proceedings of the sixth conference on Computer systems*, 2011, pp. 301–314.

[20] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[21] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[22] J. Deng, Y. Wang, J. Li, C. Shang, H. Liu, S. Rajasekaran, and C. Ding, "Tag: Gradient attack on transformer-based language models," *arXiv preprint arXiv:2103.06819*, 2021.

[23] A. Y. Ding, E. Peltonen, T. Meuser, A. Aral, C. Becker, S. Dustdar, T. Hiessl, D. Kranzlmüller, M. Liyanage, S. Maghsudi *et al.*, "Roadmap for edge ai: A dagstuhl perspective," pp. 28–33, 2022.

[24] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," *arXiv preprint arXiv:1802.04208*, 2018.

[25] H. Du, Z. Li, D. Niyato, J. Kang, Z. Xiong, H. Huang, and S. Mao, "Generative ai-aided optimization for ai-generated content (aigc) services in edge networks," *arXiv preprint arXiv:2303.13052*, 2023.

[26] H. Du, Z. Li, D. Niyato, J. Kang, Z. Xiong, D. I. Kim *et al.*, "Enabling ai-generated content (aigc) services in wireless edge networks," *arXiv preprint arXiv:2301.03220*, 2023.

[27] Y. Du, Z. Liu, J. Li, and W. X. Zhao, "A survey of vision-language pre-trained models," *arXiv preprint arXiv:2202.10936*, 2022.

[28] S. Duan, D. Wang, J. Ren, F. Lyu, Y. Zhang, H. Wu, and X. Shen, "Distributed artificial intelligence empowered by end-edge-cloud computing: A survey," *IEEE Communications Surveys & Tutorials*, 2022.

[29] I. A. Elgendy and R. Yadav, "Survey on mobile edge-cloud computing: A taxonomy on computation offloading approaches," *Security and Privacy Preserving for IoT and 5G Networks: Techniques, Challenges, and New Directions*, pp. 117–158, 2022.

[30] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," *arXiv preprint arXiv:1902.08710*, 2019.

[31] M. Erol-Kantarci and S. Sukhmani, "Caching and computing at the edge for mobile augmented reality and virtual reality (ar/vr) in 5g," in *Ad Hoc Networks: 9th International Conference, AdHocNets 2017, Niagara Falls, ON, Canada, September 28–29, 2017, Proceedings*. Springer, 2018, pp. 169–177.

[32] W. Fedus, I. Goodfellow, and A. M. Dai, "Maskgan: better text generation via filling in the_," *arXiv preprint arXiv:1801.07736*, 2018.

[33] C. Feng, P. Han, X. Zhang, B. Yang, Y. Liu, and L. Guo, "Computation offloading in mobile edge computing networks: A survey," *Journal of Network and Computer Applications*, p. 103366, 2022.

[34] F. Firouzi, B. Farahani, and A. Marinšek, "The convergence and interplay of edge, fog, and cloud in the ai-driven internet of things (iot)," *Information Systems*, vol. 107, p. 101840, 2022.

[35] D. Foster, *Generative deep learning: teaching machines to paint, write, compose, and play*. O'Reilly Media, 2019.

[36] R. Gao, X. Hou, J. Qin, J. Chen, L. Liu, F. Zhu, Z. Zhang, and L. Shao, "Zero-vae-gan: Generating unseen features for generalized

and transductive zero-shot learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 3665–3680, 2020.

[37] S. Garfinkel, *Architects of the information society: 35 years of the Laboratory for Computer Science at MIT*. MIT press, 1999.

[38] L. Girin, S. Leglaive, X. Bie, J. Diard, T. Hueber, and X. Alameda-Pineda, "Dynamical variational autoencoders: A comprehensive review," *arXiv preprint arXiv:2008.12595*, 2020.

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[40] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[41] R. Gozalo-Brizuela and E. C. Garrido-Merchan, "Chatgpt is not all you need. a state of the art review of large generative ai models," *arXiv preprint arXiv:2301.04655*, 2023.

[42] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *International conference on machine learning*. PMLR, 2015, pp. 1462–1471.

[43] S. Gu, J. Bao, D. Chen, and F. Wen, "Giqa: Generated image quality assessment," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 2020, pp. 369–385.

[44] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE transactions on knowledge and data engineering*, 2021.

[45] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 01, pp. 87–110, jan 2023.

[46] G. Harshvardhan, M. K. Gourisaria, M. Pandey, and S. S. Rautaray, "A comprehensive survey and analysis of generative models in machine learning," *Computer Science Review*, vol. 38, p. 100285, 2020.

[47] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, "Fog computing for energy-efficient data offloading of iot applications in industrial sensor networks," *IEEE Sensors Journal*, vol. 22, no. 9, pp. 8663–8671, 2022.

[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[49] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial networks and their variants work: An overview," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–43, 2019.

[50] H. Hua, Y. Li, T. Wang, N. Dong, W. Li, and J. Cao, "Edge computing with artificial intelligence: A machine learning perspective," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[51] S. A. Huda and S. Moh, "Survey on computation offloading in uav-enabled mobile edge computing," *Journal of Network and Computer Applications*, p. 103341, 2022.

[52] V. Hung, A. Gonzalez, and R. Demara, "Towards a context-based dialog management layer for expert systems," in *2009 International Conference on Information, Process, and Knowledge Management*. IEEE, 2009, pp. 60–65.

[53] A. Jabbar, X. Li, and B. Omar, "A survey on generative adversarial networks: Variants, applications, and training," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–49, 2021.

[54] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 2, pp. 494–514, 2021.

[55] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[56] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, "Ammus: A survey of transformer-based pretrained models in natural language processing," *arXiv preprint arXiv:2108.05542*, 2021.

[57] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[58] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1759–1799, 2021.

[59] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[60] S. Kim, S.-g. Lee, J. Song, J. Kim, and S. Yoon, "Flowavenet: A generative flow for raw audio," *arXiv preprint arXiv:1811.02155*, 2018.

[61] M. R. King, "The future of ai in medicine: a perspective from a chatbot," *Annals of Biomedical Engineering*, vol. 51, no. 2, pp. 291–295, 2023.

[62] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[63] H. Ko, D. Y. Lee, S. Cho, and A. C. Bovik, "Quality prediction on deep generative images," *IEEE Transactions on Image Processing*, vol. 29, pp. 5964–5979, 2020.

[64] S.-W. Ko, S. J. Kim, H. Jung, and S. W. Choi, "Computation offloading and service caching for mobile edge computing under personalized service preference," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 6568–6583, 2022.

[65] Y. Kumar, A. Koul, and C. Singh, "A deep learning approaches in text-to-speech system: A systematic review and recent research perspective," *Multimedia Tools and Applications*, vol. 82, no. 10, pp. 15 171–15 197, 2023.

[66] C.-C. J. Kuo and A. M. Madni, "Green learning: Introduction, examples and outlook," *Journal of Visual Communication and Image Representation*, p. 103685, 2022.

[67] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*. PMLR, 2016, pp. 1558–1566.

[68] H. Lee, U. Ullah, J.-S. Lee, B. Jeong, and H.-C. Choi, "A brief survey of text driven image generation and maniulation," in *2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2021, pp. 1–4.

[69] X. Lei, W. Wang, and C.-C. J. Kuo, "Genhop: An image generation method based on successive subspace learning," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2022, pp. 3314–3318.

[70] X. Lei, G. Zhao, and C.-C. J. Kuo, "NITES: A non-parametric interpretable texture synthesis method," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2020, pp. 1698–1706.

[71] X. Lei, G. Zhao, K. Zhang, and C.-C. J. Kuo, "Tghop: an explainable, efficient, and lightweight method for texture generation," *APSIPA Transactions on Signal and Information Processing*, vol. 10, 2021.

[72] F. Li, J. Qin, and W. X. Zheng, "Distributed q-learning-based online optimization algorithm for unit commitment and dispatch in smart grid," *IEEE transactions on cybernetics*, vol. 50, no. 9, pp. 4146–4156, 2019.

[73] W. Y. B. Lim, Z. Xiong, D. Niyato, X. Cao, C. Miao, S. Sun, and Q. Yang, "Realizing the metaverse with edge intelligence: A match made in heaven," *IEEE Wireless Communications*, 2022.

[74] K.-Y. Lin and G. Wang, "Hallucinated-iqa: No-reference image quality assessment via adversarial learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 732–741.

[75] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.

[76] Y. Lin, H. Du, D. Niyato, J. Nie, J. Zhang, Y. Cheng, and Z. Yang, "Blockchain-aided secure semantic communication for ai-generated content in metaverse," *IEEE Open Journal of the Computer Society*, vol. 4, pp. 72–83, 2023.

[77] D. Liu, X. Chen, Z. Zhou, and Q. Ling, "Hiertrain: Fast hierarchical edge ai learning with hybrid parallelism in mobile-edge-cloud computing," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 634–645, 2020.

[78] Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan, and Z. He, "A survey of visual transformers," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[79] B. Luo, R. Y. Lau, C. Li, and Y.-W. Si, "A critical review of state-of-the-art chatbot designs and applications," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 1, p. e1434, 2022.

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

[81] D. Martin, A. Serrano, A. W. Bergman, G. Wetzstein, and B. Masia, "Scangan360: A generative model of realistic scanpaths for 360 images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 5, pp. 2003–2013, 2022.

[82] P. McEnroe, S. Wang, and M. Liyanage, "A survey on the convergence of edge computing and ai for uavs: Opportunities and challenges," *IEEE Internet of Things Journal*, 2022.

[83] M. Mirka, M. France-Pillois, G. Sassatelli, and A. Gamatié, "A generative ai for heterogeneous network-on-chip design space pruning," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 1135–1138.

[84] M. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine learning at the network edge: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–37, 2021.

[85] P. Nema, M. Khapra, A. Laha, and B. Ravindran, "Diversity driven attention model for query-based abstractive summarization," *arXiv preprint arXiv:1704.08300*, 2017.

[86] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[87] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," *arXiv preprint arXiv:2212.08751*, 2022.

[88] A. Oussidi and A. Elhassouny, "Deep generative models: Survey," in *2018 International conference on intelligent systems and computer vision (ISCV)*. IEEE, 2018, pp. 1–8.

[89] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (gans): A survey," *IEEE access*, vol. 7, pp. 36 322–36 333, 2019.

[90] R. B. Parikh, S. Teeple, and A. S. Navathe, "Addressing bias in artificial intelligence in health care," *Jama*, vol. 322, no. 24, pp. 2377–2378, 2019.

[91] S. Parikh, D. Dave, R. Patel, and N. Doshi, "Security and privacy issues in cloud, fog and edge computing," *Procedia Computer Science*, vol. 160, pp. 734–739, 2019.

[92] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," *arXiv preprint arXiv:2104.10350*, 2021.

[93] X. Qi and C. Liu, "Enabling deep learning on iot edge: Approaches and evaluation," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 367–372.

[94] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Mirrorgan: Learning text-to-image generation by redescription," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1505–1514.

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

[96] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[97] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[98] J. Ratican, J. Hutson, and A. Wright, "A proposed meta-reality immersive development pipeline: Generative ai models and extended reality (xr) content for the metaverse," *Journal of Intelligent Learning Systems and Applications*, vol. 15, 2023.

[99] J. Ren, Y. Guo, D. Zhang, Q. Liu, and Y. Zhang, "Distributed and efficient object detection in edge computing: Challenges and solutions," *IEEE Network*, vol. 32, no. 6, pp. 137–143, 2018.

[100] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv preprint arXiv:2006.04558*, 2020.

[101] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *Advances in neural information processing systems*, vol. 32, 2019.

[102] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[103] D. Roselli, J. Matthews, and N. Talagala, "Managing bias in ai," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 539–544.

[104] L. Rui, S. Yang, S. Chen, Y. Yang, and Z. Gao, "Smart network maintenance in an edge cloud computing environment: An adaptive model compression algorithm based on model pruning and model clustering," *IEEE Transactions on Network and Service Management*, 2022.

[105] T. Safavi, C. Belth, L. Faber, D. Mottin, E. Müller, and D. Koutra, "Personalized knowledge graph summarization: From the cloud to your pocket," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 528–537.

[106] E. Samikwa, A. Di Maio, and T. Braun, "Ares: Adaptive resource-aware split learning for internet of things," *Computer Networks*, vol. 218, p. 109380, 2022.

[107] J. Selva, A. S. Johansen, S. Escalera, K. Nasrollahi, T. B. Moeslund, and A. Clapés, "Video transformers: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[108] J. Shi, Y. Liu, P. Zhou, and L. Sun, "Badgpt: Exploring security vulnerabilities of chatgpt via backdoor attacks to instructgpt," *arXiv preprint arXiv:2304.12298*, 2023.

[109] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[110] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-efficient edge ai: Algorithms and systems," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2167–2191, 2020.

[111] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti *et al.*, "Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model," *arXiv preprint arXiv:2201.11990*, 2022.

[112] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.

[113] J. Surbiryala and C. Rong, "Cloud computing: History and overview," in *2019 IEEE Cloud Summit*. IEEE, 2019, pp. 1–7.

[114] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014.

[115] M. Suzuki and Y. Matsuo, "A survey of multimodal deep generative models," *Advanced Robotics*, vol. 36, no. 5-6, pp. 261–278, 2022.

[116] Y. X. Tan, C. P. Lee, M. Neo, K. M. Lim, J. Y. Lim, and A. Alqahtani, "Recent advances in text-to-image synthesis: Approaches, datasets and future research prospects," *IEEE Access*, 2023.

[117] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Alpaca: A strong, replicable instruction-following model," *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, vol. 3, no. 6, p. 7, 2023.

[118] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–28, 2022.

[119] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du *et al.*, "Lamda: Language models for dialog applications," *arXiv preprint arXiv:2201.08239*, 2022.

[120] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," *arXiv preprint arXiv:1711.01558*, 2017.

[121] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[122] S. Tuli, N. Basumatary, and R. Buyya, "Edgelens: Deep learning based object detection in integrated iot, fog and cloud computing environments," in *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*. IEEE, 2019, pp. 496–502.

[123] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[124] B. Wang, C. Zhang, C. Wei, and H. Li, "A focused study on sequence length for dialogue summarization," *arXiv preprint arXiv:2209.11910*, 2022.

[125] C. Wang, X. Liu, and D. Song, "Language models are open knowledge graphs," *arXiv preprint arXiv:2010.11967*, 2020.

[126] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[127] Y. Wang, Z. Mei, Q. Zhou, I. Katsavounidis, and C.-C. J. Kuo, "Green image codec: a lightweight learning-based image coding method," in *Applications of Digital Image Processing XLV*, vol. 12226. SPIE, 2022, pp. 70–75.

[128] Y.-C. Wang, X. Ge, B. Wang, and C.-C. J. Kuo, "Greenkgc: A lightweight knowledge graph completion method," *arXiv preprint arXiv:2208.09137*, 2022.

[129] Z. Wang, Q. She, and T. E. Ward, "Generative adversarial networks in computer vision: A survey and taxonomy," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[130] C. Wei, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, "An overview on language models: Recent developments and outlook," *arXiv preprint arXiv:2303.05759*, 2023.

[131] R. Wei, C. Garcia, A. El-Sayed, V. Peterson, and A. Mahmood, "Variations in variational autoencoders-a comparative evaluation," *Ieee Access*, vol. 8, pp. 153 651–153 670, 2020.

[132] H. Wu, X. Lyu, and H. Tian, "Online optimization of wireless powered mobile-edge computing for heterogeneous industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9880–9892, 2019.

[133] J. Wu, W. Gan, Z. Chen, S. Wan, and H. Lin, "Ai-generated content (aigc): A survey," *arXiv preprint arXiv:2304.06632*, 2023.

[134] Z. Xiao, Z. Xia, H. Zheng, B. Y. Zhao, and J. Jiang, "Towards performance clarity of edge video analytics," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2021, pp. 148–164.

[135] H. Xu, Y. Wang, Y. Wang, J. Li, B. Liu, and Y. Han, "Acg-engine: An inference accelerator for content generative neural networks," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–7.

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

[137] M. Xu, D. Niyato, J. Chen, H. Zhang, J. Kang, Z. Xiong, S. Mao, and Z. Han, "Generative ai-empowered simulation for autonomous driving in vehicular mixed reality metaverses," *arXiv preprint arXiv:2302.08418*, 2023.

[138] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[139] J. Yao, S. Zhang, Y. Yao, F. Wang, J. Ma, J. Zhang, Y. Chu, L. Ji, K. Jia, T. Shen *et al.*, "Edge-cloud polarization and collaboration: A comprehensive survey for ai," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[140] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "Qa-gnn: Reasoning with language models and knowledge graphs for question answering," *arXiv preprint arXiv:2104.06378*, 2021.

[141] J. Zhai, S. Zhang, J. Chen, and Q. He, "Autoencoder and its various variants," in *2018 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2018, pp. 415–419.

[142] C. Zhang, C. Zhang, S. Zheng, Y. Qiao, C. Li, M. Zhang, S. K. Dam, C. M. Thwal, Y. L. Tun, L. L. Huy *et al.*, "A complete survey on generative ai (aigc): Is chatgpt from gpt-4 to gpt-5 all you need?" *arXiv preprint arXiv:2303.11717*, 2023.

[143] C. Zhang, C. Zhang, M. Zhang, and I. S. Kweon, "Text-to-image diffusion model in generative ai: A survey," *arXiv preprint arXiv:2303.07909*, 2023.

[144] C. Zhang, C. Zhang, S. Zheng, M. Zhang, M. Qamar, S.-H. Bae, and I. S. Kweon, "A survey on audio diffusion models: Text to speech synthesis and enhancement in generative ai," *arXiv preprint arXiv:2303.13336*, vol. 2, 2023.

[145] H. Zhang, H. Song, S. Li, M. Zhou, and D. Song, "A survey of controllable text generation using transformer-based pre-trained language models," *arXiv preprint arXiv:2201.05337*, 2022.

[146] J. Zhang and D. Tao, "Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7789–7817, 2020.

[147] W. Zhang, J. Chen, Y. Zhang, and D. Raychaudhuri, "Towards efficient edge cloud augmentation for virtual reality mmogs," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, 2017, pp. 1–14.

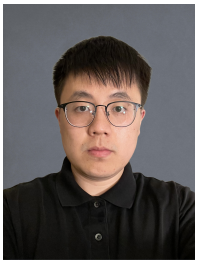
[148] Z. Zhang, C. Li, W. Sun, X. Liu, X. Min, and G. Zhai, "A perceptual quality assessment exploration for aigc images," *arXiv preprint arXiv:2303.12618*, 2023.

[149] Z. Zhang, A. Pinto, V. Turina, F. Esposito, and I. Matta, "Privacy and efficiency of communications in federated split learning," *IEEE Transactions on Big Data*, 2023.

[150] M. Zhao, J. Li, F. Tang, S. Asif, and Y. Zhu, "Learning based massive data offloading in the iov: Routing based on pre-rlga," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2330–2340, 2022.

YUN-CHENG WANG (Student Member, IEEE) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2018. He received his M.S. degree in 2020 and is working towards Ph.D. degree in electrical and computer engineering with the Ming-Hsieh Department of Electrical and Computer Engineering, University of Southern California. His research interest includes knowledge graph embedding, natural language processing.

JINTANG XUE (Student Member, IEEE) received the B.S. degree in communication engineering from Shanghai University, Shanghai, China, in 2021. He received his M.S. degree in 2023 and is working towards Ph.D. degree in electrical and computer engineering with the Ming-Hsieh Department of Electrical and Computer Engineering, University of Southern California. His research interest includes natural language processing.



CHENGWEI WEI (Student Member, IEEE) received the B.S. degree in automation from Central South University, Changsha, in 2018. He received his M.S. degree in 2020 and is working towards Ph.D. degree in electrical and computer engineering with the Ming-Hsieh Department of Electrical and Computer Engineering, University of Southern California. His research interest includes natural language processing, representation learning, and image processing.

C.-C. Jay Kuo (Fellow, IEEE) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1980, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1985 and 1987, respectively. He is the holder of the William M. Hogue Professorship in Electrical and Computer Engineering, a Distinguished Professor of Electrical and Computer Engineering and Computer Science, and the Director of the USC Multimedia Communication Laboratory (MCL) at the University of Southern California. He is the co-author of about 340 journal papers, 1000 conference papers, and 15 books. His research interests include multimedia and green computing. He is a fellow of the National Academy of Inventors, the American Association for the Advancement of Science, and the International Society for Optical Engineers, and the Association for Computing Machinery.