Toward Federated Large Language Models: Motivations, Methods, and Future Directions

Yujun Cheng, Weiting Zhang, *Member, IEEE*, Zhewei Zhang, Chuan Zhang, *Member, IEEE*, Shengjin Wang, *Senior Member, IEEE*, and Shiwen Mao, *Fellow, IEEE*

***Abstract*—Large Language Models (LLMs), such as LLaMA and GPT-4, have transformed the paradigm of natural language comprehension and generation. Despite their impressive performance, these models still face certain challenges, including the need for extensive data, high computational resources, and privacy concerns related to their data sources. Recently, Federated Learning (FL) has surfaced as a cooperative AI methodology that enables AI training across distributed computation entities while maintaining decentralized data. Integrating FL with LLMs presents an encouraging solution for privacy-preserving and collaborative LLM learning across multiple end-users, thus addressing the aforementioned challenges. In this paper, we provide an exhaustive review of federated Large Language Models, starting from an overview of the latest progress in FL and LLMs, and proceeding to a discourse on their motivation and challenges for integration. We then conduct a thorough review of the existing federated LLM research from the perspective of the entire lifespan, from pre-training to fine-tuning and practical applications. Moreover, we address the threats and issues arising from this integration, shedding light on the delicate balance between privacy and robustness, and introduce existing approaches and potential strategies for enhancing federated LLM privacy and resilience. Finally, we conclude this survey by outlining promising avenues for future research in this emerging field.**

***Index Terms*—Federated learning, large language model, foundation model, privacy.**

I. Introduction

IN THE past few years, the domain of Artificial Intelligence (AI) [1], [2], [3], [4], [5], [6], [7], [8] has experienced a paradigm shift with the advent of Foundation Models (FMs), prominently represented by Large Language Models (LLMs). LLMs, including GPT series [9], [10], [11], PaLM [12], and LLaMA [13], boast billions of parameters and have attracted considerable interest owing to their outstanding performance across a wide spectrum of AI tasks, such as text generation [14], contextual awareness [15], and even planning and reasoning [16]. Based on these AI tasks, LLMs have paved the way for a plethora of applications in diverse fields. They provide technical assistance not only to areas directly linked to language processing (e.g., search engines [17], service support [18], and multi-language translation [19], [20]), but also prove beneficial in broader contexts such as code generation [21], chatbot [22], finance [23], and legal consultation [24]. However, while LLMs have achieved impressive success in various domains, these models necessitate substantial amounts of high-quality data and significant computational resources, which result in substantial costs for the training and utilization of LLMs. Moreover, LLMs typically rely on extensive public datasets for training. To enhance their performance in specific domains, they need to incorporate data from private entities, such as hospitals and banks. However, these highly sensitive data pose privacy challenges that hinder further improvements in LLMs unless well addressed.

Received 15 May 2024; revised 31 August 2024 and 18 October 2024; accepted 17 November 2024. Date of publication 21 November 2024; date of current version 14 August 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62394321 and Grant 62201029, and in part by the State Key Development Program in 14th Five Year under Grant 2021QY1702. *(Corresponding author: Weiting Zhang.)*

Yujun Cheng, Zhewei Zhang, and Shengjin Wang are with the Department of Electronic Information Engineering, Tsinghua University, Beijing 100084, China (e-mail: yjcheng@tsinghua.edu.cn; demonmikalis@126.com; wgsgj@tsinghua.edu.cn).

Weiting Zhang is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: wtzhang@bjtu.edu.cn).

Chuan Zhang is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: chuanz@bit.edu.cn).

Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smao@ieee.org).

Digital Object Identifier 10.1109/COMST.2024.3503680

Nomenclature

Acronyms	*Definitions*
AI	Artificial Intelligence
ML	Machine Learning
FL	Federated Learning
LM	Language Model
LLM	Large Language Model
NLP	Natural Language Processing
GPT	Generative Pre-Training Transformer
i.i.d	Independently and Identically Distributed
PFM	Pretrained Foundation Model
IoT	Internet of Things
UAV	Unmanned Aerial Vehicle
HFL	Horizontal Federated Learning
VFL	Vertical Federated Learning
TFL	Transfer Federated Learning
CFL	Centralized Federated Learning
DFL	Decentralized Federated Learning
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
BERT	Bidirectional Encoder Representation from Transformers

RLHF	Reinforcement Learning with Human Feedback
KD	Knowledge Distillation
GPU	Graphics Processing Unit
PEFT	Parameter-Efficient Fine-tuning
LoRA	Low-Rank Adaptation of Large Language Models
FLOP	Floating Point Operations
SMPC	Secure Multi-Party Computation
DP	Differential Privacy
HE	Homomorphic Encryption
IP	Intellectual Property
GAN	Generative Adversarial Network
ViTs	Vision Transformers Models
CLIP	Contrastive Language-Image Pre-training

In the field of machine learning, Federated Learning (FL) has recently gained traction as an innovative framework that enables intelligent learning systems while maintaining data privacy [25], [26], [27], [28]. FL is a decentralized artificial intelligence strategy that facilitates model training across numerous devices, coordinated by a central server, without the need to exchange the actual data. Typically, the process begins with the central server, or the aggregator, initializing a global model with certain learning parameters. Each participating device, referred to as a worker, retrieves the latest model from the aggregator, applies its own data to update the model, and then transmits this updated model back to the aggregator. The aggregator then amalgamates these updates from all workers to refine the global model. By harnessing the computing power of distributed workers, significant computational resources can be saved for the centralized server. Additionally, since the data remains local, the data transmission cost associated with centralized training is reduced, and user privacy risks are minimized.

With these distinctive advantages, FL can address various challenges faced by LLM systems, spanning from pre-training to deployment, as previously discussed. LLM can also assist FL in synthetic data generation to mitigate the non-i.i.d (non-independent and identically distributed) data challenge within FL. Therefore, the integration of LLMs and FL, namely federated LLM, is emerging as a prominent and trending topic [29], [30], [31]. FL leverages distributed data sources to efficiently supply LLMs with a vast amount of data, which is one of the primary requirements for LLMs. Additionally, the use of distributed data sources helps circumvent the communication overhead typically associated with centralizing distributed datasets. FL's capacity for reliable privacy protection during distributed training enables access to high-quality, private domain data beyond publicly available datasets. For instance, medical data from hospitals can be harnessed to address knowledge gaps in LLMs within specific domains, thereby enhancing model accuracy and generalizability. Furthermore, the distributed computing nature of FL allows for the exploitation of computational resources from numerous edge devices. This exploitation can, to some extent, alleviate the immense computational power required by large models. However, integrating FL with LLMs introduces new and unexplored challenges. These challenges include heterogeneity in federated LLM training, as well as new privacy and security concerns arising from this amalgamation. For example, the integration of FL with LLMs raises significant concerns regarding data leakage and model inversion attacks. Adversaries may exploit the gradients shared during the training process to infer sensitive information about the underlying data. Therefore, it is imperative to develop advanced cryptographic techniques and differential privacy methods to safeguard against such threats. Furthermore, the amalgamation of FL and LLMs necessitates stringent access control and authentication protocols to prevent unauthorized access and ensure the confidentiality of the training data. The implementation of Secure Multi-Party Computation (SMPC) and homomorphic encryption can provide viable solutions to these privacy and security challenges. To examine the current state of research and address the encountered challenges, this article will conduct a comprehensive survey of the federated LLM domain.

A. Related Reviews and Our Contributions

Prompted by the latest progress in FL and LLM, a number of overviews on the corresponding studies have emerged. For instance, Khan et al. [32] review the recent advances of FL for enabling IoT applications. They propose a comprehensive set of evaluation metrics, such as sparsification, robustness, quantization, scalability, security, and privacy, to rigorously evaluate the recent progress. Furthermore, they establish a systematically structured classification for understanding FL in IoT networks. Similarly, Nguyen et al. [33] investigate FL for a range of crucial IoT services and explore related works in IoT applications. Other studies such as [34], [35], [36], [37] have also presented the fundamental principles of FL. and the taxonomy of recent work under various scenarios. In addition, Lyu et al. [38] delve into the privacy and robustness challenges in FL, offering a thorough classification of FL threats along with the respective protective measures and outlining prospective avenues for future research. Several other similar works [14], [39] also provide a detailed taxonomy from the FL privacy and security perspective.

On the other hand, with the rise of LLMs, there has been a surge in survey works based on LLMs. For instance, Zhou et al. [40] present an extensive survey of the latest developments, existing challenges, and future prospects for Pretrained Foundation Models, with a particular focus on LLMs among diverse data types. Xi et al. [41] delve into the realm of LLM-based agents, exploring current research and future prospects in this domain. They introduce a general conceptual framework for LLM-based agents, comprising three essential components: brain, perception, and action, which is adaptable to various applications. Wang et al. [42] offer a comprehensive overview of LLM alignment technologies, discussing from three major perspectives, including data collection, training methodologies, and model evaluation. This serves as a crucial guide for individuals interested in comprehending and progressing the alignment of LLMs to more effectively meet human-centric tasks and expectations. In [43], the authors conduct a thorough examination of the security and privacy issues associated with LLMs concerning both training data and application-based risks across various domains. The

review includes an assessment of the vulnerabilities inherent to LLMs, an investigation into the emerging security and privacy attacks targeting these models, and a comprehensive evaluation of potential defense strategies. Several other works [44], [45], [46], [47] also provide detailed introductions to various aspects of LLMs.

While FL and LLMs have been studied extensively in isolation within the existing literature, as far as we know, only a selected number of studies have conducted an in-depth analysis specifically focused on the system architecture of federated LLMs, a categorization of existing federated LLM works, and federated LLM applications in various scenarios. For instance, Yu et al. [48] and Chen et al. [49] emphasize the potential benefits and challenges of FL throughout the lifecycle of LLMs, including stages of pre-training, fine-tuning, and application. They also delve into future research directions, aiming to facilitate the creation of more personalized and context-sensitive models, all while prioritizing data privacy protection. Similar work has been proposed by Zhuang et al. [50], aiming to understand of the synergistic relationship between FL and LLM. Their study highlights the motivations, challenges, and future directions in more detail. Nonetheless, these studies have merely offered a cursory overview of the fundamental notions and challenges, without conducting a comprehensive review of all the current relevant research. These limitations inspire us to carry out a more thorough survey of the integration of FL and LLM. The principal contributions of this paper are highlighted as follows:

1) We present a state-of-the-art survey on the topic of federated LLM, beginning with an introduction to the basic system concept and recent advances in FL and LLM. We also engage in an in-depth discussion about the motivation behind integrating FL and LLM, delving into how this union can foster innovation and enhance efficiency across both fields.
2) We conduct a comprehensive investigation and analysis of the existing work, spanning from pre-training and fine-tuning to application. Various detailed topics, such as data construction, initialization, and research related to heterogeneity, are thoroughly reviewed. Additionally, we provide taxonomy tables summarizing the key technical aspects and contributions of each proposed approach for federated LLM.
3) Furthermore, we undertake a thorough review from the perspectives of privacy and robustness. We have curated a list of potential threats to privacy and security that federated LLMs may encounter, along with a detailed analysis of corresponding defense strategies. Furthermore, we offer taxonomy tables summarizing these research efforts, providing a clear overview of the field's current state and challenges.
4) We identify several critical research challenges and outline prospective research directions that aim to boost the performance and utility of federated LLMs.

B. Structure of the Survey

The structure of this paper is systematically outlined in Fig. 1. Section II offers a primer on the foundational concepts of the system and delves into the latest developments in FL and LLMs. It particularly explores the core mechanisms of FL, the various types of federated methodologies, and the progression, architecture, and taxonomy of LLMs. Section III examines the impetus and obstacles associated with merging FL with LLMs. Section IV provides a comprehensive survey and categorization of the current work on federated LLMs, from pre-training and fine-tuning to application, and meticulously sorts through different subtopics in detail. Section V addresses the work on federated LLMs from the perspectives of privacy and robustness, investigates potential privacy and security threats that may arise, and presents an in-depth analysis of the respective countermeasures. Section VI discusses potential future directions aimed at enhancing the performance of federated LLMs. Section VII summarizes the survey.

II. Background

A. Federated Learning

FL has gained significant attention across multiple research fields, leading to a proliferation of studies on FL. Since its inception in 2016 [51], FL has been a game-changer for a multitude of intelligent Internet of Things (IoT) applications. It has offered groundbreaking AI solutions, capitalizing on its distributed framework and privacy-conscious features. FL can be described as a distributed approach to ML. In this framework, clients independently conduct training on their datasets and update a collective global model at a central server without exposing their individual data. This process enables devices to benefit from a comprehensive model while preserving data privacy, as they periodically contribute to the enhancement of the global model by sharing their model updates.

1) Definition of Federated Learning: In the standard FL framework, it is assumed that there are N participating clients, denoted as $\{C_1, C_2, \ldots, C_N\}$. Each client C_n possesses a private dataset $D_n = \{x_i^n, y_i^n\}_{i=1}^{K_n}$ with $|x^n| = K_n$ and $K = \sum_{n=1}^{K} N_k$. In addition, client C_n typically possesses a learned local network model or an initialized model, represented by $f(\theta_n)$. Thus, $f(x_n, \theta_n)$ represents the predicted result of the private sample x_n based on the local model θ_k. Conventional ML frameworks are typically built on a larger centralized dataset, denoted as $D_{central} = D_1 \cup D_2 \cup \cdots \cup D_N$, by directly combining the private datasets of each client. This merged dataset is subsequently employed to train a model with better performance, symbolized by $\theta_{central}$. Despite the limitations imposed by data silos and privacy concerns, traditional centralized learning approaches are impractical for use in real-world contexts where privacy is paramount. As a remedy, FL allows each participant, C_n, to jointly train the models without revealing the private data D_n to other clients $C_{n_0}(n \neq n_0)$. As shown in Fig. 2, the typical FL procedure includes the following key stages:

1) *System Initialization.* The server receives the task requirements and target application, then establishes learning parameters, for instance, learning rates and communication rounds. In addition, the server chooses a set of clients, denoted as $\{C_1, C_2, \ldots, C_K\}$, to participate and set up the initial global model θ_{global}^1 in the

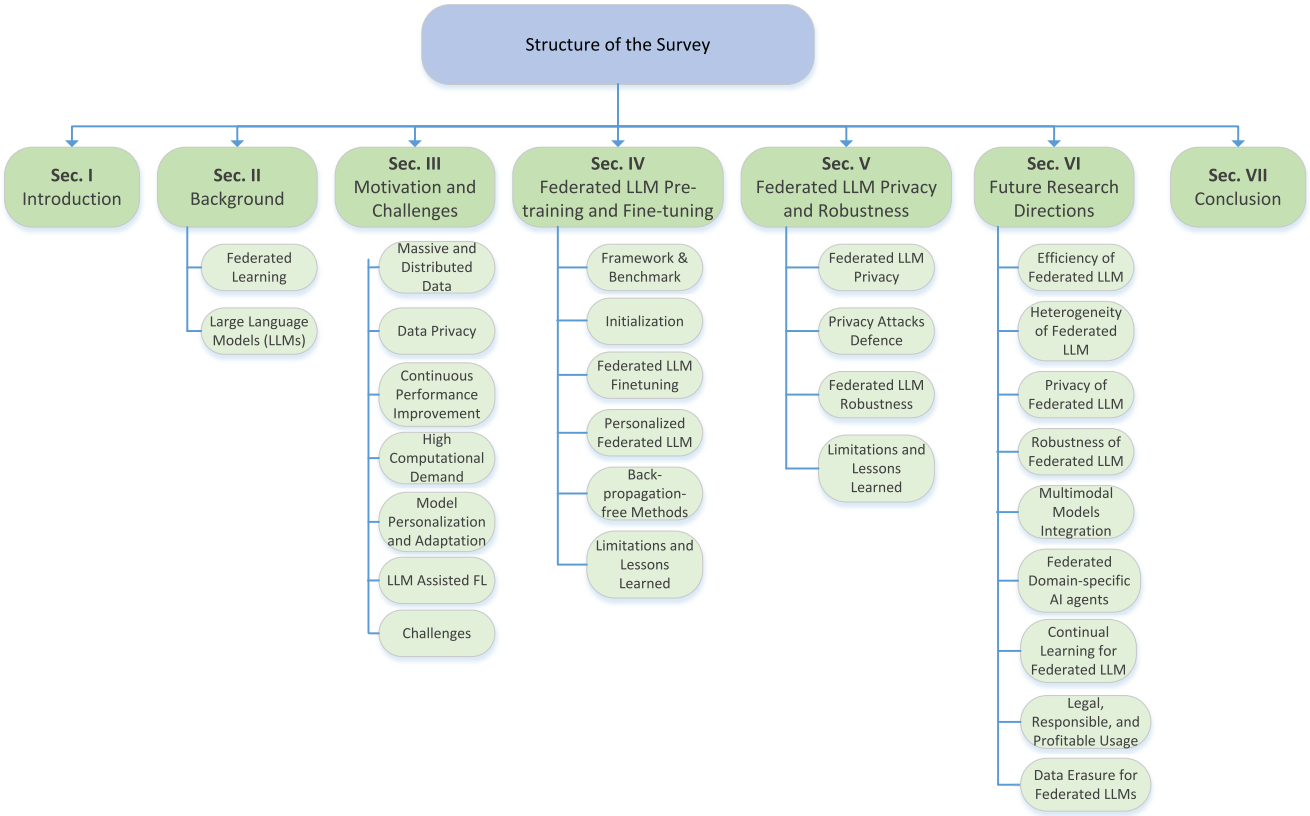


Fig. 1. Organization of this paper.

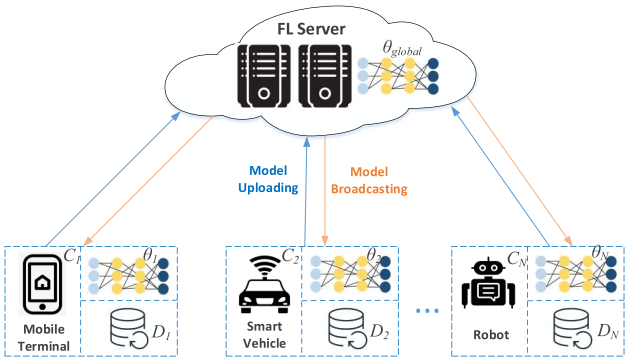


Fig. 2. A typical FL procedure considering N number of participants.

first round. Subsequently, it distributes the present global model θ_{global}^{t-1} to all the clients involved. This serves as the initialization of local models $\theta_1^{t-1}, \theta_2^{t-1}, \ldots, \theta_N^{t-1}$.

2) *Local Training and Update.* Each client C_n employs its private dataset D_n to execute local model updates in the following manner:

$$\theta_n^t \leftarrow \theta_n^{t-1} - \alpha \nabla_\theta \mathcal{L}_n \Big(f(x^n, \theta_n^{t-1}), y^n \Big), \tag{1}$$

where α denotes the learning rate and $\mathcal{L}_n(\cdot)$ denotes the calculated loss for each client n. The loss function can differ among various FL approaches [28]. Subsequently, each client, labeled as n, sends its computed update θ_n^t to the central server.

3) *Model Aggregation and Global Update.* Upon receiving the local models $\{\theta_1^t, \theta_2^t, \ldots, \theta_N^t\}$ from the participants, the FL server conducts an aggregation process and subsequently generates an updated version of the global model θ_{global}^t by

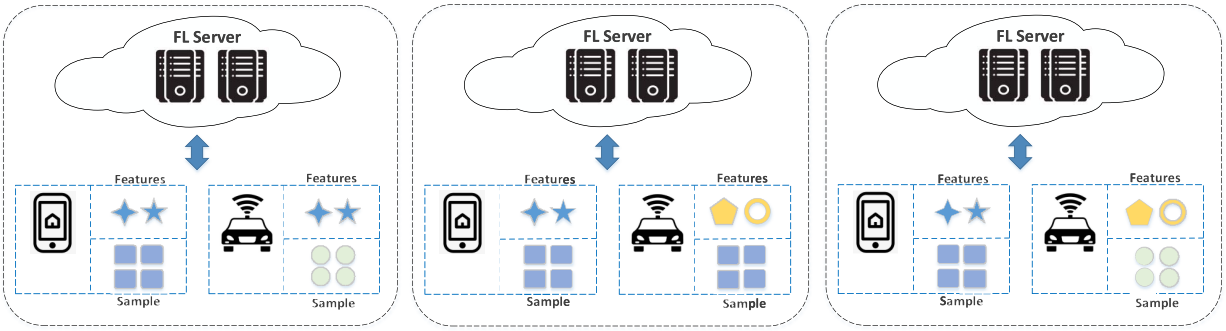
$$\theta_{global}^t \leftarrow \frac{1}{\sum_{n \in N} |D_n|} \sum_{n=1}^{N} |D_n| \theta_n. \tag{2}$$

After aggregating the models, the server disseminates the latest global update θ_{global}^t to every client. This is intended to enhance the local models in the ensuing training iteration. The FL procedure is reiterated until the global loss function reaches convergence or attains a predetermined accuracy benchmark.

2) Taxonomy of FL Frameworks: Recent research in FL [33], [35], [52] has made significant advances. Typically, FL can be categorized according to federation scale, partitioning sample and networking structure. In this subsection, we conduct a detailed study of each individual instance of categorization.

Depending on the nature of the data distribution across clients, FL is typically classified into three distinct categories, including Horizontal Federated Learning (HFL) [53], Vertical Federated Learning (VFL) [54], and Transfer Federated Learning (TFL) [55], as summarized in Fig. 3.

HFL is the predominant technique utilized in FL. It encompasses the federation of samples and is optimal in situations characterized by a high degree of feature overlap but



(a) Horizontal FL (b) Vertical FL (c) FTL

Fig. 3. FL with different data organization strategies. The main difference between these methods is whether they have the same or different types of samples and features.

minimal node overlap. Such conditions are commonly linked with cross-device scenarios. In an HFL framework, every participant independently develops their AI model, which results in a local update. To ensure security and privacy, these local updates can be concealed through methods like encryption or application of differential privacy. Following this, a central server consolidates all the local updates received from the clients to formulate a new, comprehensive global update. This global update is then disseminated back to the clients, facilitating the subsequent phase of local training. This iterative cycle persists until the model's loss function stabilizes or attains a predetermined accuracy threshold.

In contrast, VFL and TFL present intricate challenges in their implementation and integration within various application contexts due to their distinct approaches to data structuring. VFL is designed to facilitate the collaborative learning of a shared AI model among a network of clients that possess a common set of samples but disparate feature sets. It employs an entity alignment method to amalgamate intersecting samples from these clients, which are then used to collectively train a unified AI model. Security during this process is bolstered by the use of encryption protocols. Conversely, TFL is the strategy of choice in scenarios where there is a scant overlap of features and samples across nodes. It involves the transformation of features from heterogeneous feature spaces into a unified format, enabling the training of a model with data compiled from numerous clients. The server responsible for aggregation then updates the model based on the weights received from the participants' learning processes. TFL's primary goal is to develop tailored models that are effective for specific use cases, particularly when data availability is limited, thereby representing another crucial facet of data organization within FL strategies.

Networking Structure: From the standpoint of network architecture, FL can be bifurcated into two subcategories: Centralized Federated Learning (CFL) and Decentralized Federated Learning (DFL).

Currently, CFL is the most widely adopted framework in FL. This structure involves a central server coordinating with numerous clients to implement an FL model. During each iteration of training, clients individually train their models on local data and then forward the updated parameters to the central server. The server employs a specialized algorithm for aggregating these parameters, such as Federated Averaging (FedAvg) [51], to produce a global model. This global model is then circulated among all clients for further training rounds. However, the centralized design of CFL can lead to issues like a single point of failure, dependency on trust, and server bottlenecks. These are common challenges in systems where a central server plays a pivotal role. In contrast to the CFL system, DFL presents a server-less paradigm of FL. This approach underscores the benefits of adopting a peer-to-peer model for delivery and aggregation, which is independent of a central trusted server. Rather than exclusively interacting with a central server, participants in a Decentralized FL system can fully exploit the network bandwidth by utilizing the network connections among themselves. This peer-to-peer communication allows for a more distributed and potentially more resilient system. These modern attributes of DFL make it compatible with peer-to-peer communication technologies, including blockchain [56], [57], to establish decentralized FL networks.

B. Large Language Models (LLMs)

LLMs are typically sophisticated language models, distinguished by their extensive parameter sizes and exceptional learning capabilities. The self-attention module in the Transformer [58] acts as the fundamental building block for these LLMs, aiding in language modeling tasks. Furthermore, these LLMs necessitate substantial computational resources and diverse datasets for model pretraining. After the training phase, LLMs require finetuning to meet specific downstream requirements, including performance, speed, and confidentiality. This section will offer a comprehensive introduction to the background of LLMs concerning the aforementioned aspects.

1) LLM Background and Definition: Language models (LMs) [59], [60] are computational models designed to comprehend and generate human language. Here, we focus on generative language models that generate text in an autoregressive manner. These models predict the next token in a sequence by calculating the probability distribution conditioned on the preceding tokens, which unfolds as follows:

$$\mathbb{P}(w) = \mathbb{P}(w_1) \cdot \mathbb{P}(w_2|w_1) \cdots \mathbb{P}(w_T|w_1, \ldots, w_{T-1}), \quad (3)$$

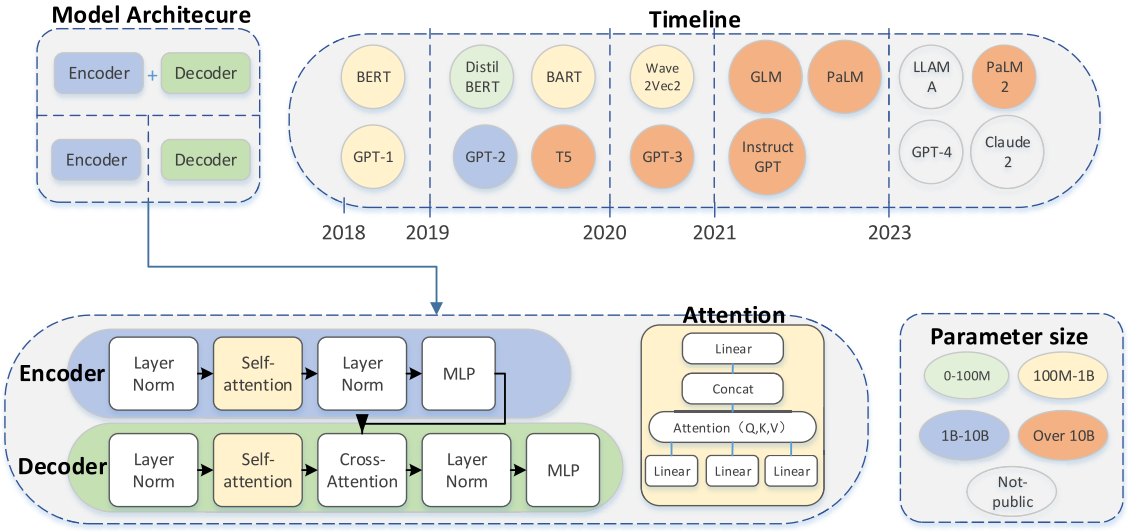


Fig. 4. The model architecture, the Transformer structure, and evolution of LLMs.

where $\{w_1 \cdots w_T\}$ denotes a text sequence of $T = |w|$ tokens, and t is the current position. $\mathbb{P}(w_t|w_1, \ldots, w_{t-1})$ with $t = 1, \ldots, T$, is the probability the LM outputs on the token w_t given the previous $t-1$ tokens. However, traditional LMs usually encounter several challenges, such as dealing with rare or unseen words, mitigating the issue of overfitting, and the difficulty in capturing complex linguistic phenomena.

LLM have made great progress in solving the problems that traditional LMs faced before, utilizing huge amounts of high-quality data and having large amounts of parameters. The fundamental structure that has driven the recent advancements of LLMs is the Transformer model. This model, first introduced in 2017, has outperformed the traditional Recurrent Neural Network (RNN) architecture and become the preferred model for machine translation tasks. Transformer models are more suitable for parallel computing than RNN, which makes them to train faster and handle larger datasets. The Transformer architecture is built around two key structures, namely an **encoder** and a **decoder**. The encoder is constructed from multiple identical layers, each containing a multi-head attention mechanism and a feed-forward neural network. These layers work in unison to process the input sequence, extracting its features layer by layer, and culminate in a final output that is passed on to the decoder. Similarly, the decoder is comprised of several identical layers, each equipped with a multi-head attention mechanism and a feed-forward neural network. However, the decoder layers also include an additional encoder-decoder attention mechanism, which focuses on the input sequence while decoding. At the core of the Transformer lies the mechanism of self-attention. The fundamental formula for the attention mechanism is given by:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (4)$$

where Q, K, and V represent Query, Key, and Value vectors, respectively, while d_k denotes the dimension of the key. The self-attention mechanism quantifies the relevance of each term within a sentence when forecasting a particular word. It computes a weighted aggregate of the values for all terms in the sentence, with the weights determined by the resemblance of each term to the word under prediction.

As discussed, all modern LLMs are based on the Transformer architecture. This design allows these models to handle up to trillions of parameters. Fig. 4 illustrates a thorough overview of LLMs, including the model architecture, the evolutionary trajectory of the LLMs, and the Transformer structure.

2) Training and Fine-Tuning of LLMs: The pre-training process of large-scale language models involves numerous crucial steps that are essential for their effective development. This procedure usually begins with the collection and preprocessing of a vast amount of text data from diverse sources. The assembled dataset is viewed as the fundamental building block for the training of LLMs. Table I shows several examples of datasets used by typical models. During the training process, unsupervised learning techniques are employed, wherein the model learns to predict the subsequent word in a sequence based on the context that precedes it. This task is commonly referred to as language modeling. LLMs utilize sophisticated neural network architectures, usually Transformers, which allow them to capture intricate language patterns and dependencies. The primary training objective is to optimize the model's parameters to maximize the likelihood of generating the correct next word within a given context. This optimization is typically achieved using an algorithm like stochastic gradient descent (SGD) or its variants, combined with backpropagation to iteratively update the model's weights. In addition, pre-training for LLMs necessitates significant computational power. For example, GPT-3-175B [10] model utilizes 3.14×10^{23} flops and LLaMa-70B [13] requires 1.7×10^6 hours of GPU processing. Consequently, the strategic allocation of computational assets is crucial for the streamlined pre-training of these language models.

On the other hand, the fine-tuning techniques for LLMs can be classified into the following three categories:

TABLE I
DATASETS UTILIZED BY SEVERAL TYPICAL LLM MODELS

LLM	Datasets
GPT-3 [10]	CommonCrawl [61], WebText2 [62], Wikipedia [63], Books1 , Books2
LLaMA [13]	CommonCrawl , Wikipedia , C4 [64], Books, Github, Arxiv, StackExchange
T5 [64]	C4 , WebText, Wikipedia, RealNews
PaLM [12]	Social Media, Wikipedia, Books, Github, Webpages
CodeGen [65]	BIGQUERY [66], BIGPYTHON, the Pile
GLM [67]	BooksCorpus [68], Wikipedia

Supervised Fine-Tuning (SFT): The fundamental principle of SFT [69] involves refining the model through supervised learning after extensive pre-training, thereby enhancing its proficiency in conforming to the unique demands of a specific task. During the SFT procedure, it's crucial to have a labeled dataset specific to the given task, featuring input texts paired with their respective labels. A subset of SFT, known as instruction tuning, is commonly employed in the fine-tuning stages of LLMs. This method further trains LLMs using a dataset of both instruction and output, with the goal of bolstering the models' ability to comprehend and execute human instructions, thus improving their functionality and controllability.

Alignment Tuning: Alignment tuning is imperative. To tackle the challenges posed by LLMs acting outside human intentions [70]. LLMs undergo initial training with vast amounts of varied data sourced from the Internet. Despite preprocessing efforts, it remains challenging to fully eliminate biased or detrimental content from the extensive datasets used in training. While LLMs have shown remarkable abilities in a range of language processing tasks, they can sometimes produce results that stray from what humans might intend, such as creating inaccurate information or biased and misleading expressions [71]. A prominent strategy for alignment tuning involves the use of Reinforcement Learning with Human Feedback (RLHF) [69], which incorporates human-generated feedback to develop a model that guides the reinforcement learning process.

Parameter-efficient Tuning: LLMs such as ChatGPT [11] are constantly increasing in scale. However, for most researchers, full fine-tuning on consumer-grade hardware is infeasible and impractical. PEFT's objective is to meet both computational and memory demands in LLM finetuning. This approach entails fine-tuning merely a select few or additional model parameters, while the bulk of the pre-trained parameters remain unchanged, thus significantly diminishing the need for computational and storage resources. Notably, cutting-edge techniques in parameter-efficient tuning have achieved results on par with comprehensive fine-tuning. Among the prominent methods of parameter-efficient tuning are Low-Rank Adaptation (LoRA) [72], Prefix Tuning [73], and P-Tuning [74], [75]. These techniques enable effective tuning of models, even in settings with limited resources, offering practicality and efficiency for real world applications.

III. MOTIVATION AND CHALLENGES

In this section, we will elaborate on the motivation and challenges of combining LLMs with FL. Current FL techniques face several challenges when dealing with LLMs, including model complexity and communication overhead. Due to the large number of parameters in LLMs, traditional FL methods incur high communication costs during model updates and parameter synchronization. Additionally, FL has limitations in handling heterogeneous data and devices, which can affect model performance and convergence speed. In particular, we will investigate how FL can address the current issues in the LLM training and application processes, and vice versa, how LLMs can support FL in various aspects.

A. Massive and Distributed Nature of LLM Training Data

LLMs are typically pretrained using massive and high-quality data to achieve an astounding performance. For instance, GPT-3 used 45TB of text data for training, while Meta LLaMA-2 used 20 trillion tokens for training. However, such high-quality data is projected to be exhausted within five years [76]. Moreover, platforms that previously offered free public data, such as Twitter, have begun charging substantial fees for accessing their data [50]. In addition, utilizing these public data may also involve legal and copyright-related complications. It will become increasingly difficult to train better-performing LLMs using public datasets.

Conversely, a substantial volume of data remains accessible within private domains, spanning a wide array of personal and corporate sources. However, aggregating these distributed private datasets for centralized training would not only necessitate intricate data integration efforts but also pose potential privacy risks. Considering both model performance and efficiency, FL stands out as a promising solution. By directly utilizing private data for model training, it addresses the challenges posed by privacy and data distribution across various domains. Besides, by training with FL, LLMs can access a wider range of data for optimization tasks such as fine-tuning, prompt tuning, and pre-training. The enhanced data access facilitates the creation of more accurate and efficient AI systems, better tailored to meet the needs of users across a wide range of application scenarios.

B. Data Privacy

Data privacy is a crucial concern in LLM training and application, given the massive and distributed data used. In FL, the server does not need raw data for training. The server and clients only exchange intermediate information in model training, such as model weights or gradient updates. This core idea ensures that sensitive data stays local and is not leaked. Therefore, FL reduces the risk of exposing sensitive

user information to external third parties and enhances data privacy in LLM training and application.

C. Continuous Performance Improvement With Updating Data

Another data-related challenge is the necessity of keeping LLMs updated with the latest knowledge. Data in real-world scenarios are constantly growing even during a short time. For instance, common applications such as drones and mobile robots are constantly generating new data over time [77]. Maintaining the relevance and timeliness of LLMs becomes challenging, particularly when dealing with distributed data. The dynamic nature of information across various sources necessitates continuous adaptation and updates to ensure LLMs remain accurate and effective. FL provides a solution by enabling continuous adaptation and enhancement of models by utilizing distributed and heterogeneous data sources. For instance, LLMs can be deployed in a federated manner, where local models undergo additional fine-tuning based on the data specific to that locality. Rather than transmitting local data, only the updates to the model are transmitted back to the central server. This allows the global model to be progressively enhanced based on user data, without ever directly accessing that data.

D. High Computational Demand for LLM Training

As discussed in Section II, LLMs have a huge number of model parameters. Hence, training large-scale LLMs demands considerable computational resources. This poses a challenge for separate entities lacking the necessary framework or capabilities to independently conduct LLM training. FL facilitates a collective training approach, permitting entities to combine their computational capabilities, which in turn, decentralizes the training workload and alleviates the burden on any single entity. On the other hand, FL usually involves multiple local clients with heterogeneous computing resources in the entire life-span process, and it can adjust the original model according to the node's computing capability, so that even clients with low computing power can also join in the LLM training and fine-tuning process.

E. Model Personalization and Adaptation

With the rapid development of LLMs, an increasing number of large models are evolving towards specific domains. These application scenarios not only involve different specific domain knowledge, but also have certain constraints on the computing power and hardware requirements of clients. Therefore, due to the decentralized nature of FL, it can provide users with personalized and adaptive LLM services by training on diverse, user-generated data. Another important issue is the bias in LLM training. In FL, the models learn from various users, which diversifies the data and knowledge that LLMs use. This helps the models better understand the nuances and complexities of real-world scenarios, and leads to more informed and less biased decisions for different tasks and domains, which contribute to bias reduction in LLM systems.

F. LLM-Assisted FL

While FL can address numerous challenges associated with LLMs, LLMs can reciprocally offer substantial support to classic FL. For instance, a common issue in classic FL is client drift, which arises from the heterogeneity in private data distribution among clients. To improve FL performance, recent research suggests incorporating data gathered from public domains, such as the Internet, into the FL process. The success of methods that utilize public data is largely dependent on the quality of the gathered public data. To overcome the constraints associated with public data, approaches based on synthetic data for FL have been developed. LLMs, which are pre-trained on wide-ranging datasets, have powerful fitting capabilities of data distribution. This enables them to create synthetic data that faithfully reflects the varied and intricate nature of real-life data scenarios.

Additionally, LLMs can effectively address the issue of suboptimal performance in FL through a process known as knowledge distillation [78], [79]. Knowledge distillation is a technique where the LLM, functioning as the "teacher," imparts its knowledge to streamline the training of a more basic "student" model within the FL framework. The LLM usually employs knowledge distillation to refine and condense the student model. Subsequently, each participant in the FL network utilizes this distilled student model to bolster their local training efforts. The transfer of insights from the LLM to the smaller model elevates the latter's performance and ability to generalize, addressing the challenges posed by scarce or unevenly distributed data. This approach allows for more efficient and effective learning within the FL system.

G. Challenges

The integration of LLMs with FL provides many benefits as we explained above, yet it also inherits certain fundamental challenges from the existing LLM and FL paradigms. In this context, we will delve into these challenges, with a specific emphasis on architectural design, privacy and security issues.

Computational and Communication Resource Issues: In conventional Federated Learning FL, memory, communication, and computation costs are critical factors that significantly impact performance. These challenges become even more pronounced when integrating LLMs due to their substantial size and complexity. One of the core principles of FL is the frequent exchange of model updates between clients and the central server [80], [81], [82]. When dealing with LLMs, the volume of data that needs to be transmitted and synchronized across multiple clients is immense. This results in significant communication overhead, which can lead to increased latency and reduced overall system performance. The high communication cost can also be a barrier in scenarios with limited bandwidth or unstable network connections. Training LLMs is computationally intensive, requiring substantial processing power and energy consumption. In an FL setting, this computational burden is distributed across multiple clients, many of which may not have the necessary computational resources to handle such demanding tasks. This can lead to uneven training progress and suboptimal model

performance, as some clients may struggle to keep up with the computational demands. To address these resource challenges, several strategies can be employed. Techniques such as pruning [83], [84], quantization [85], [86], and knowledge distillation [87], [88] can be used to reduce the size of LLMs, making them more manageable for FL environments. These methods help in decreasing memory and computation requirements without significantly compromising model performance. Instead of training LLMs from scratch, fine-tuning pre-trained models [89], [90] on specific tasks can significantly reduce the computational and communication costs. This approach leverages the knowledge already embedded in the pre-trained models, requiring fewer resources for adaptation to new tasks.

Synchronization and Coordination: In FL, clients independently train their local models and periodically synchronize with a central server. However, when dealing with LLMs, the large number of parameters and the complexity of these models exacerbate these issues. One of the primary challenges is the timeliness of model updates. Due to the enormous size of LLMs, clients need to transmit a vast amount of update data to the central server after local training. This large-scale data transmission not only increases communication overhead but also leads to delays and potential staleness in updates. Stale updates can slow down the convergence of the global model and reduce its accuracy. To address this, efficient synchronization mechanisms are required. One approach is synchronous aggregation [91], where the server waits for updates from all clients before performing a global update. However, this can lead to increased latency, especially if some clients have slower network connections or lower computational power. On the other hand, asynchronous aggregation [92], [93], [94] allows the server to update the global model as soon as it receives updates from any client. While this can reduce latency, it introduces the risk of incorporating stale updates, which can degrade the model's performance. To mitigate this, techniques such as staleness-aware aggregation [95], [96] can be employed, where the server assigns different weights to updates based on their timeliness, giving more importance to recent updates. Straggler clients [97], [98] are another issue that needs to be addressed. These are slower clients that delay the synchronization process. One approach to mitigate this is to set a deadline for updates, after which the server proceeds with the available updates, ignoring the stragglers. Another approach is to use partial aggregation [99], [100], where the server aggregates updates from a subset of clients, ensuring that the synchronization process is not held up by a few slow clients.

Heterogeneities: In FL, data is distributed across multiple clients, each of which may possess vastly different data in terms of quantity, quality, and distribution. The non-IID nature of this data can lead to model divergence and suboptimal performance when training LLMs, namely a phenomenon known as data heterogeneity [101], [102]. Additionally, clients in a federated setting can range from powerful servers to resource-constrained edge devices, each with varying computational power, memory capacity, and network conditions. This variability can result in uneven training progress, as some clients may struggle to meet the computational demands of training LLMs, which referred to as system heterogeneity [103], [104]. To address these heterogeneities, techniques such as personalized federated learning [105], [106], [107] can be employed. Personalized federated learning aims to create models tailored to the specific data distribution and computational capabilities of each client, rather than relying on a single global model that may not perform well for all clients. Methods such as meta-learning [108] and multi-task learning [109] can be explored to enable effective personalization.

Privacy and Security Issues: LLMs need to deal with various potential attacks and biases, such as adversarial examples [110], [111], backdoor attacks [112], [113], poisoning attacks [114], [115], model stealing [116], [117], etc. These challenges lead to issues concerning the robustness and security of LLMs, which can impact their reliability and trustworthiness. For instance, LLMs are susceptible to word embedding poisoning due to noisy perturbations. Studies [118] indicate that even modifying a single word embedding vector enables an adversary to subtly manipulate the model, leading to abnormal responses to specific trigger words. Furthermore, the impact of word embedding poisoning attacks in federated networks is substantial; even a small number of compromised clients can significantly degrade the global model. In federated LLM systems deployed over wireless networks, adversarial jamming emerges as a practical threat, corrupting sensitive word embeddings during transmission.

The challenges mentioned above is by no means a complete list, as separate research on LLMs and FL has already addressed a substantial number of issues [119], [120], [121], [122]. However, this paper mainly focuses on the research that integrates LLMs with FL more cohesively, and the works that are biased towards only LLM or FL are out of our scope. This paper will introduce and discuss these challenges in detail and depth, and review the existing major relevant works.

IV. LLM Pre-Training and Fine-Tuning With FL: State-of-the-Art

In this section, we conduct a comprehensive review of the literature on LLM that utilize FL for pre-training and fine-tuning, representing the state-of-the-art in this field. Specifically, we survey the existing federated LLM systems and organize our discussion around five key aspects: framework & benchmark, data and model initialization, federated LLM finetuning, personalized federated LLM, backpropagation-free methods.

A. Framework & Benchmark

The pioneering effort in large-scale FL systems was made by Google, where FL was employed to enhance next-word prediction [123] and query suggestion [124] for Gboard applications. Following this, various innovative FL systems have been developed to accommodate different FL scenarios, such as TFF [125], FedLab [126], Felicitas [127], IBM FL [128], Paddle-FL [129]. Recently, there has been notable advancement in the creation of FL infrastructures and standards specifically tailored for LLMs. These frameworks, usually

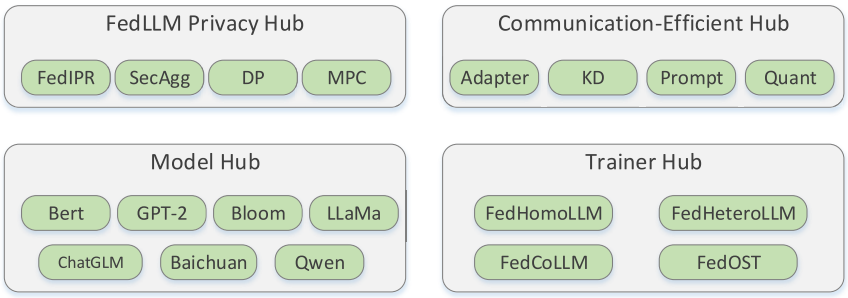


Fig. 5. Basic components of FATE-LLM.

coupled with LLMs, provide a complete set of APIs that facilitate various effective FL algorithms, encompassing most or all aspects of the lifecycle of federated LLMs.

FedLLM [130], [131], FATE-LLM [132] and FederatedScope-LLM (FS-LLM) [133] are three notable works in the field of federated LLMs for enterprise level applications. They incorporate parameter-efficient fine-tuning techniques to enhance training efficiency and reduce communication overhead. Additionally, these systems implement robust privacy-preserving mechanisms to ensure the confidentiality and integrity of data during both training and inference phases. By facilitating cross-domain collaboration, these frameworks allow diverse organizations to jointly optimize model performance while addressing the challenges of limited computational resources and data heterogeneity. FedML presents FedLLM [130], which enables an MLOps-based training pipeline to construct the enterprise's own LLM on private data. FedLLM can perform training in both centralized and geo-distributed GPU clusters, as well as in a FL fashion for data silos. For a particular siloed GPU cluster, FedLLM utilizes existing open-source LLMs and well-known frameworks for local training. Moreover, for efficient training, FedLLM supports parameter-efficient training methods such as LoRA.

FATE-LLM (Fig. 5), built upon FATE (Federated AI Technology Enabler), aims to simplify FL for LLMs [132]. Specifically, FATE-LLM enables FL for both homogeneous and heterogeneous LLMs; it also boosts the training efficiency of Federated LLM by adopting parameter-efficient fine-tuning methods, such as LoRA and P-Tuning-v2. Furthermore, FATE-LLM protects the intellectual property of LLMs using a federated protection approach and preserves data privacy during training and inference by applying privacy-preserving mechanisms. This holistic approach strives to optimize the performance of LLMs while adhering to high standards of data security and privacy.

FS-LLM [133] aims to represent an all-encompassing toolkit designed for the federated refinement of LLMs. This toolkit establishes a seamless benchmarking system from start to finish, streamlining dataset preparation, the execution or simulation of federated refinement, and the assessment of performance in federated LLM refinement tailored for various capability demonstrations. It offers a suite of ready-to-deploy federated PEFT algorithms and adaptable interfaces for programming, which pave the way for future enhancements to LLM functionalities in FL environments, ensuring minimal communication and computational overhead, and even enabling operation without full model access. Moreover, it integrates a range of swift and economical operators for optimizing LLMs under resource constraints, along with modular sub-routines that support cross-disciplinary research (such as personalized FL applications of LLMs).

OpenFedLLM [134] presents an innovative framework for contemporary LLMs that promotes cooperative and secure training on underutilized distributed private datasets. It utilizes FL to collectively enhance a shared model while maintaining the privacy of raw data. The framework is streamlined, unified, and designed to be user-friendly. Additionally, it integrates federated instruction tuning to refine LLMs' ability to follow instructions, federated value alignment to align LLMs with human ethics, and includes 7 key FL algorithms. OpenFedLLM also facilitates the training of LLMs across multiple fields, encompassing 8 distinct training datasets, and offers thorough evaluations with over 30 evaluation metrics.

In addition, the academic community also has a variety of benchmarks and implementations for federated LLMs. FedIT [135] is recognized as a pioneering initiative that utilizes FL for instruction tuning of LLMs. This research illustrates that FedIT can address the constraints of traditional instruction tuning by harnessing the varied sets of instructions from users within the FL framework. This is particularly effective in a cross-device FL environment with a client base numbering in the billions. Additionally, it offers an in-depth analysis of the diversity present in FL instruction tuning. Utilizing the GPT-4 auto-evaluation technique, the study validates the efficacy of FedIT in enhancing the quality of responses through the use of a broad spectrum of instructions.

Woisetschlager et al. [29] explore the present and future capabilities of edge computing for FL with LLMs, by contrasting these systems with a data-center GPU. They show the possibility for improvement and the next steps towards achieving higher computational efficiency at the edge. Specifically, this study fine-tunes the FLAN-T5 model family, adopting FL for a text summarization task. It provides a micro-level hardware benchmark, compared the model FLOP utilization to a state-of-the-art GPU used in a data center, and examined the network utilization in realistic conditions.

Zhao et al. [136] introduce a methodology that integrates privacy-preserving technologies including FL, emulator-based tuning with PEFT strategies and differential privacy (DP). The paper details specialized parameter-efficient methods for federated environments, designed to minimize communication costs while maintaining model efficacy. Additionally, the implementation of DP safeguards against compromising individual data privacy during statistical analysis.

There are also several other modalities of federated foundation models training/fine-tuning works, such as Flower [137], which supports fine-tuning of Whisper for the downstream task of keyword spotting in a federated way. The experiments also benchmark the new Raspberry Pi 5, with regard to not only training times but also the time taken to pre-process the dataset partitions. FedCLIP, as presented in [138], introduces a lightweight adapter module for the CLIP. These streamlined adapters are capable of harnessing the extensive knowledge of pretrained models, thereby guaranteeing that the models remain flexible and suitable for client-specific applications. Gao et al. [139] conduct a study on the convergence of self-supervised learning and FL, with an emphasis on speech representation leveraging the wav2vec 2.0 framework [140]. The researchers provide a pioneering, systematic exploration into the practicality involved in developing speech models within FL contexts, examining the subject through the lenses of algorithmic processes, hardware capabilities, and systemic boundaries.

B. Data and Model Initialization

The processing of original data and the initial setup of models are crucial factors influencing the efficacy of FL. Among the current FL literature, neural networks are usually initialized with random weights. However, in centralized learning, it is common to use model initialization with weights pre-trained on large-scale datasets, as this has been proven [141], [142] to enhance accuracy, generalizability, robustness, etc. Training from random weights is also more challenging for LLMs. Some recent works have investigated whether model pre-training is suitable for FL and the impact of model initialization (whether random or pre-trained) on the performance of federated optimization techniques. These studies [143], [144] show that initiating from a pre-trained LLM can notably diminish the disparity between Independent and Identically Distributed (IID) and non-IID data settings for clients. Furthermore, when using a pre-trained model as the initial point, the number of local epochs per round can be greatly decreased without degrading the final accuracy. These results indicate that pre-training effectively closes the gap between FL and centralized learning.

One of the challenges for LLM is how to group such massive datasets. Dataset Grouper library [145] facilitates the formation of extensive, group-structured datasets, such as those used in federated settings for Large LLMs. Its three primary benefits include its ability to manage single-group datasets that exceed memory capacity, its adaptable approach to choosing the foundational dataset and defining partitions, and its framework-agnostic nature. The generation of synthetic data using LLMs is another intriguing topic. It arises from the challenges in obtaining specific public datasets and the lack of clear guidelines for their acquisition. To address the scarcity of public data, researchers have explored FL approaches that leverage synthetic data. In these methods, a generative model is fine-tuned through knowledge distillation (KD), and synthetic data is generated by this model in a staggered manner during the federated training cycles. However, these methods have stability and security issues, which can be effectively solved by LLMs due to their strong generative performance. GPT-FL [146] utilizes generative pre-training approaches to create a variety of synthetic datasets. Such synthetic information is employed to enhance a server-based downstream model, which undergoes further refinement using confidential data from clients within the conventional FL structure. Experimental results indicate that GPT-FL outperforms contemporary FL techniques in aspects such as accuracy of model testing, efficiency of communication, and efficiency in sampling across clients. Another typical work is proposed by Wang et al. [147], which propose an in depth study into the utilization of extensive public datasets and LLMs to enhance the DP based training of mobile FL models. Their objective is to strike a better balance between privacy and utility by employing knowledge distillation methods. Additionally, they introduce an innovative distribution matching technique, backed by theoretical analysis, to select public data that closely resembles the distribution of private data, thereby effectively boosting the efficiency of training with public datasets.

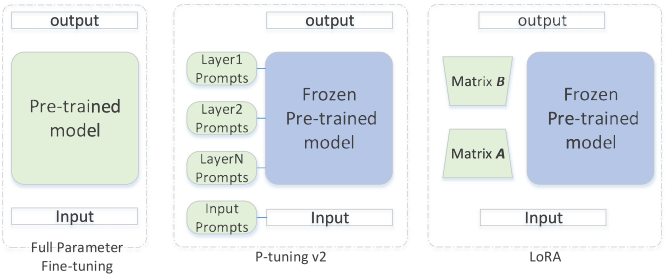


Fig. 6. Several typical fine-tuning methods, including full parameter fine-tuning, P-tuning v2, and LoRA. The weights that require fine-tuning are highlighted within the green boxes.

C. Federated LLM Fine-Tuning

The process of federated fine-tuning LLMs focuses on adapting a pre-trained LLM to achieve specific objectives. Due to the considerable computational resources and data volume requirements for pre-training LLMs, most of the federated LLM work concentrates on enhancing the fine-tuning phase's effectiveness and efficiency. For instance, efforts are made to reduce the communication and computational overhead of federated LLM fine-tuning, tackle the challenges of non-IID data and personalized requirements across various clients, and support a broader range of downstream tasks.

1) Federated LLM With PEFT: As outlined in Section II-B, PEFT is a technique to tailor LLMs to particular downstream tasks. Several typical PEFT methods are shown in Fig. 6. The methodology entails freezing the core architecture of LLMs

while modifying a minimal number of extra parameters. The objective of PEFT is to decrease the duration of training and the expenses associated with communication, thereby addressing a fundamental obstacle in FL. Here we only introduce the typical PEFT-related works, and the works on personalized FL based on PEFT will be described in detail in the next subsection.

LP-FL [148] integrates few-shot prompt learning from LLMs with efficient communication and federating techniques. It empowers federated clients to assign soft labels to unlabeled data, using the gradually updated knowledge from the global model. By employing the LoRA technique, this method enables the generation of concise learnable parameters and promotes the federation of global models with minimal overhead.

Malaviya et al. [149] demonstrate that the PEFT approach is capable of diminishing communication overheads without compromising the efficacy of the model in FL environments. In a range of real-world applications, acquiring data specific to the intended downstream task can prove challenging; however, procuring data from analogous tasks is often more feasible. The empirical evidence regarding the task-level applicability of PEFT within FL frameworks indicate substantial zero-shot learning capabilities of the model on the intended tasks, provided that the source data is derived from a closely related task.

FedTune [150] reveals that Transformers, once fine-tuned, exhibit superior efficacy in FL scenarios. The study highlights that a streamlined fine-tuning process not only accelerates the convergence but also minimizes communication costs. Delving into the specifics, the authors conducted an extensive empirical analysis on three distinct tuning strategiesmodifying inputs, integrating additional modules, and altering the core architecture, which utilizing two categories of pre-trained models, namely vision-language and vision models. The findings underscore that the integration of pre-trained models within FL significantly enhances accuracy of the model by effectively addressing the problem of overfitting.

FedPETuning [151] investigates the potential of PEFT methods in LLMs and establish a federated benchmark for four key PEFT approaches. Specifically, it pioneers a thorough empirical investigation into the tuning techniques of prominent pretrained LMs within an FL framework, encompassing aspects such as privacy threats, performance metrics, and analysis under resource limitations. The extensive experimental data corroborate that FedPETuning is adept at safeguarding privacy while maintaining commendable model performance, all the while reducing the demand for substantial resources.

FedPrompt [152] examines the application of prompt tuning through a model split aggregation approach within FL. It reveals that this method markedly diminishes the communication overhead while only slightly impacting accuracy across both IID and Non-IID data distributions. Additionally, the study includes tests of backdoor attacks via data poisoning on FedPrompt, with results showing a minimal success rate for the attacks and an inability to implant a backdoor successfully, thereby affirming FedPrompt's resilience.

PROMPTFL [153] introduces a shift from traditional federated model training to federated prompt training. This approach encourages participants in a federated network to focus on training prompts rather than a communal model. This strategy aims to harness the capabilities of LLMs for efficient global aggregation and effective local training, even with limited data. The key advantage of PROMPTFL lies in its requirement to update only the prompts, not the entire model, thus significantly speeding up both local training and global aggregation processes. Moreover, an LLM that has been trained on extensive datasets can offer robust adaptability for diverse tasks of distributed users through the utilization of trained soft prompts.

SplitLoRA [154] is a split FL LLM fine-tuning framework, which is constructed upon the split FL paradigm, integrating the benefits of parallel training from FL and model partitioning from split learning. This integration significantly enhances training efficiency by delegating the primary training workload to a server through model partitioning. This approach involves the exchange of activations and their gradients with smaller data sizes, rather than transmitting the entire LLM. Notably, SplitLoRA represents the first open-source benchmark for SL LLM fine-tuning, establishing a foundational platform for future research endeavors aimed at advancing split FL LLM fine-tuning methodologies.

2) Fine-Tuning for Downstream Applications: LLMs must undergo the fine-tuning process in order to perform better on downstream tasks. This stage adjusts the pre-trained model's parameters using a dataset tailored to the task at hand. The purpose of fine-tuning is to shift from the broad linguistic comprehension acquired during pre-training to a focus on the particular subtleties and demands of the task in question. Some works have adopted FL to finetune the LLM for downstream tasks, with the aim of addressing the privacy and data sensitivity issues.

Riedel et al. [155] explore the application of FL for the purpose of Multilingual Protest News Detection (a binary classification task), utilizing news texts in English, Portuguese, Spanish, and Hindi. The researchers engage fine-tuning pre-trained multilingual BERT and DistilBERT models on the multilingual data, leveraging their demonstrated success in analogous NLP classification tasks within centralized learning frameworks [156], [157]. Additionally, the study assesses the performance of FL aggregation algorithms across different data partitioning scenarios.

FedTherapist [158] is a mobile application designed for mental health monitoring that leverages ongoing speech and typing activity while ensuring privacy through FL. The authors assess the efficiency and computational load of BERT and GPT-3.5 to manage the intricacies associated with training language models directly on mobile devices. Additionally, the study presents a context-aware learning technique, which is adept at harnessing the voluminous and varied text data from smartphones for detecting mental health indicators.

FedJudge [159] combines FL and LLM to address the data privacy issues raised by the centralized training of Legal LLMs, as legal data is scattered across various institutions with sensitive personal information. In particular, FedJudge

employs parameter-efficient fine-tuning methods to update only a few extra parameters during FL training. Moreover, continual learning methods are also investigated to maintain the global model's crucial parameters when training local clients to alleviate the problem of data shifts.

FedED [160] proposes a medical relation extraction model that preserves the privacy of data sources, using FL as the underlying framework. This is to tackle the challenge of handling medical texts, which often contain sensitive information that cannot be shared or copied across different domains. The model is based on BERT models as the backbone, and enables the training of a global model without exposing or transferring any private local data. To address the communication bottleneck in FL, a knowledge distillation strategy is adopted that leverages the predictions of local models aggregated to form the global model, instead of sending local parameters.

Ahmed et al. [161] propose an active learning based news article retrieval model in a semi-supervised learning scenario. This model offers the benefits of low communication overhead, high scalability, enhanced data privacy, and a temporal-aware retrieval model. The framework employs lexicon expansion, content segmentation, and temporal events to construct a BERT attention embedding query that captures the temporal dynamics of sequential news articles. To produce pseudo-labels, the partially trained models with the original labeled data are fused.

FedHumor [162] is an approach for personalized humor recognition based on FL. Recognizing the subjective nature of humor, this method seeks to tailor humor recognition to individual preferences, a task that is notably challenging for conventional models. By enhancing a pre-trained language model, FedHumor fine-tunes its processes to reflect the varied humor tastes of different users. Furthermore, it employs a strategy that adapts to this diversity, aiming to cultivate a humor recognition model that is personalized for each user within the FL paradigm.

Efficient-FedRec [163] is an FL framework for news recommendation that preserves user privacy. This framework strategically splits the model into a substantial news component hosted on the server and a compact user model shared between the server and client devices. Specifically, client devices receive the user model and news details from the server and return their individual gradient updates for collective integration. Subsequently, the server refines its universal user model utilizing these combined gradients.

FEWFEDWEIGHT [164] trains client models on isolated devices without data sharing. The framework utilizes the server's global model to create pseudo data for each client, facilitating knowledge transfer from the global model to improve the client models' few-shot learning capability. An energy-based algorithm is implemented to filter out noise by weighting the pseudo samples. Additionally, the client models' weights are adjusted according to their performance, it then dynamically consolidates the client models to refresh the global model.

FedNLP [165] is a framework designed to assess FL techniques across four prevalent NLP tasks, namely text classification, sequence tagging, QA, and seq2seq generation. This system offers a standardized platform that integrates Transformer-based LLMs, such as BERT and BART, with FL strategies, accommodating a range of non-identically distributed data situations.

In summary, we list the federated LLM fine-tuning works in the taxonomy Table II, summarizing the technical aspects as the key informations and contributions of each reference work.

D. Personalized Federated LLM

As outlined in Section III, FL encounters various types of heterogeneity issues, which require adaptive and personalized solutions to overcome these challenges. In this paper, the definition of personalized federated LLMs is inherited from conventional personalized FL and extended to federated LLMs. The goal is to address the various heterogeneity issues that may arise during the pretraining and fine-tuning processes of federated LLMs. **Statistical (Data) Heterogeneity** arises as client data often exhibit non-IID characteristics. This can lead to a scenario where a model trained on local data at a client site may outperform a global FL model trained on heterogeneous data. **System Heterogeneity** is evident as FL clients typically operate on a wide array of hardware capabilities, including differences in computational power, network bandwidth, and storage capacities, as well as disparities in operating systems, applications, and other software tools as noted by Jiang et al. [166]. This variation allows clients with advanced hardware to train more complex models, whereas those with less capable systems are restricted to simpler models. **Model Heterogeneity** is characterized by the fact that various entities often maintain distinct, proprietary model collections. The process of fine-tuning these models within the FL framework can lead to reduced training durations while also safeguarding proprietary knowledge, as discussed by Ye et al. [35]. Given the diversity of models across organizations, it is crucial to facilitate the training of personalized models that are heterogeneous in nature. These challenges are not unique to FL but are also prevalent in federated LLMs, subsequently we will delve into the specific works addressing these challenges.

Fed-PepTAO [167] aims to enable efficient and effective FL of LLMs. This work proposes a parameter-efficient prompt tuning method with an efficient and effective method to select appropriate layers of prompts for FL. Second, a new adaptive optimization method is devised to tackle the client drift problems on both the client and server sides to improve the system performance further.

Profit [168] attempts to study the trade-off between personalization (adaptation to the clients' local distributions) and robustness (avoiding catastrophic forgetting) over different FL training algorithms and different data heterogeneity levels. The study finds that in federated LLM prompt tuning, the choice of adaptive optimizer, learning rate, regularization and other parameters is crucial for achieving the personalization vs robustness trade-off.

FedDAT [169] is a fine-tuning framework tailored for heterogeneous multi-modal FL. It employs a Dual-Adapter Teacher (DAT) to address data heterogeneity by regularizing

TABLE II
TAXONOMY OF FEDERATED LLM FINE-TUNING RESEARCH

Reference	Classification	Fine-tuning Methods	Model	Main Dataset	Key Contribution
LP-FL [148]	Fine-tuning with PEFT	LoRA	BERT-Large	IMDB, Yelp	Federated fine-tuning LLMs with PEFT method with limited communications and local computational powers.
Malaviya *et al.* [149]		Adapter, prefix, LoRA, BitFit	BERT-base	GLUE	Analyze the performance of four PEFT methods under different non-i.i.d. settings in FL
FedTune [150]		Three PEFT methods	CLIP	CIFAR-10, CIFAR-100	Conduct an in-depth measurement on various parameter-efficient tuning methods in FL using different pre-trained Transformer models.
FedPETuning [151]		Adapter, prefix, LoRA, BitFit	Roberta-Base	GLUE	Benchmark to provide a holistic review of PETuning methods for PLMs under FL settings, covering privacy attacks, performance comparisons, and resource-constrained analysis
FedPrompt [152]		Prompt tuning	BERT, Roberta, T5	OffensEval, IMDB, Twitter, GLUE	Evaluate new FL prompt tuning method performance and test the model robustness to backdoor attack
PromptFL [153]		Prompt tuning	CLIP	Cal101, Flowers102,UCF101, Sun397	Prove the feasibility of the system in terms of overhead in communication, training, and inference dimensions as well as generalization and personalization ability.
SplitLoRA [154]		LoRA + split learning	GPT-2	E2E	Constructed upon the split FL paradigm, integrating the benefits of parallel training from FL and model partitioning from split learning. ability.
RIEDEL *et al.* [155]	Downstream Applications Finetuning	Full parameter	BERT, DistilBERT	CASE2021	Fine-tuning the pre-trained DistilBERT and BERT models with the partitioned protest news reports in a federated manner.
Fedtherapist [158]		LoRA	BERT, RoBERTa, LLaMa-7B	IRB	Fine-tuning a mobile mental health monitoring system that utilizes continuous speech and keyboard input in a privacy-preserving way via FL.
FedJudge [159]		LoRA	Baichuan-7B	C3VG, Lawyer LLaMA	Utilizes PEFT to fine-tune Legal LLMs efficiently and effectively during the FL training.
FedED [160]		Full parameter	BERT	2010i2b2, GAD, EU-ADR	Propose a privacy preserving medical relation extraction model based on FL.
Ahmed *et al.* [161]		Full parameter	BERT	SemEval 2019 Task 4	Present a news article retrieval model based on active learning in a semisupervised FL setting.
FedHumor [162]		-	BERT	SemEval-2020 shared Task 7	Propose the FedHumor approach for the recognition of humorous content in a personalized manner through FL.
Efficient-FedRec [163]		-	BERT	MIND Adressa	An efficient FL framework using split method for privacy preserving news recommendation.
Fewfedweight [164]		-	BART-base	Huggingface 118 tasks	Propose a FL framework for multi-task learning in the few-shot and private setting.

the local updates of the client and applying Mutual Knowledge Distillation (MKD) for efficient knowledge transfer. The experiment results indicate that the approach attains a superior convergence rate and scalability compared to existing PEFT methods.

FedRA [170] tackles the issue of diverse client capabilities in computational and communication aspects within FL. It operates by generating a random allocation matrix in each communication cycle. For clients with constrained resources, FedRA adapts a subset of the model's layers according to this matrix and refines them through LoRA. The server then gathers the refined LoRA parameters, aligns them with the existing allocation matrix, and assimilates them into the designated layers of the basic model.

FedLoRA [171] aims to address the issues of statistical, system, and model heterogeneity in LLM PEFT. It works by incorporating a small and consistent adapter into each client's heterogeneous local model. These models are trained using an iterative procedure that facilitates the transfer of global and local knowledge. The FL server then aggregates these small and uniform local adapters into a global one. This technique allows FL clients to leverage diverse local models with reduced computational and communication costs.

pFedPG [172] is designed to harness the strong representational power of LLMs to enable efficient personalization for clients with varying capabilities. It achieves this through a dual-stage optimization process that involves adapting personalized prompts at the local level and generating them at the global level. A specialized prompt generator at the server side plays a pivotal role in this process, utilizing personalized optimization trajectories to produce unique prompts tailored for each client's model. This approach ensures that each client's model is updated effectively, taking into account the specific needs and constraints of the client's environment.

Heterogeneous-LoRA [173] explores the performance trade-off of federated fine-tuning with higher and lower LoRA ranks. It deploys heterogeneous ranks across clients, aggregates the heterogeneous LoRA modules via zero-padding, and redistributes the LoRA modules heterogeneously through truncation. By combining the benefits of high-rank and low-rank LoRAs, it achieves an optimal balance, demonstrating a simple yet effective approach.

AdaFL [165] addresses the critical issue of determining the ideal depth and breadth for fine-tuning adapters, which significantly influences the speed and efficiency of training. The optimal configuration is contingent upon the specific downstream NLP tasks, the desired accuracy of the model, and the available mobile resources. AdaFL employs a gradual approach to modify the adapter configuration throughout a training session. Initially, it focuses on rapidly learning shallow knowledge by training a limited number of smaller adapters in the upper layers of the model. Subsequently, it incorporates progressively larger and deeper adapters to grasp more complex knowledge. Additionally, AdaFL continuously evaluates various adapter configurations by designating participant devices to different experimental groups.

PERADA [174] is an effective personalized FL framework that minimizes communication and computational costs while improving generalization performance, especially under test-time distribution shifts. It enhances generalization by aligning each client's personalized adapter with a global one. The global adapter in turn employs knowledge distillation to aggregate generalized information from all clients. The validity of this approach is supported by both theoretical and empirical evidences.

Besides addressing the heterogeneity and personalization issues of federated LLM with PEFT, decomposing a substantial LLM into multiple sub-models is an uncomplicated yet promising strategy for implementing practical federated LLM. FedBERT [175] employs the concepts of federated and split learning to pre-train BERT in a distributed fashion. After the global model's pre-training phase, each client has the ability to independently fine-tune their model for local NLP tasks. This method is inclusive, supporting all participants, regardless of their computational power or data volume, to partake in the pre-training process.

FEDBFPT [176] is a framework that trains selected layers of BERT in an efficient way, which lowers the computational and communication costs. It allows the training of a large global model using FL by creating small local models for each client. These local models train specific layers of the global model, which leads to less computational resource consumption and fewer weights to send. The efficiency of FEDBFPT is supported by theoretical analysis and experiments on corpora from various domains.

FEDOBD [177] is a novel framework that splits large-scale deep models into semantic blocks, and assesses block importance (rather than individual parameter importance) and selectively eliminates unimportant blocks to achieve more significant reduction of communication cost while maintaining model performance. Comprehensive experimental evaluation shows that FEDOBD surpasses state-of-the-art baselines in terms of communication cost and test accuracy.

FedPerfix [178] attempts to investigate the partial personalization of large-scale models. It conducts empirical evaluations to determine how sensitive different layers are to data distribution. The findings suggest that the self-attention layer and the classification head in a Vision Transformer (ViT) are particularly sensitive. To address this, FedPerfix employs plugins as a means to personalize the model by transferring information from the aggregated model to individual clients.

In addition, there are also a number of works that address the heterogeneity and personalization issues of federated LLM by using techniques such as model compression [179], [180] and knowledge distillation [181], [182]. For instance, RaFFM [183] introduces specialized model compression algorithms tailored for FL scenarios, such as salient parameter prioritization and high-performance subnetwork extraction. These algorithms enable dynamic scaling of given transformer-based FMs to fit heterogeneous resource constraints at the network edge during both FL's optimization and deployment stages. Fed-ET [184] is a method that utilizes ensemble knowledge transfer within a FL framework. It involves training smaller models with varied architectures on client devices and then using these models to inform the training of a larger, more comprehensive model on a central server. This approach is distinct from traditional ensemble learning because it leverages the heterogeneous data from various clients. Fed-ET incorporates a weighted consensus distillation strategy along with diversity regularization to ensure that the consensus derived from the ensemble is reliable and to improve the model's generalization capabilities by making use of diverse data sets. However, these methods have many applications of general personalized FL [135], [185], [186], but few works have adopted these methods in federated LLM so far. In the taxonomy Tab. II, the personalized federated LLM works are cataloged, summarizing the technical aspects, key information, and contributions of each cited study.

E. Back-Propagation-Free Methods

Due to the huge amount of parameters and data, the high computational cost and memory overhead of LLM training and fine-tuning are often unacceptable, even with methods such as compression, quantization, and knowledge distillation. To tackle this issue, several studies have explored backpropagation-free techniques for training and fine-tuning federated large language models. These approaches enhance LLMs without the dependency on backpropagation. The BP-free propagation algorithm obviates the necessity to store activation values during computation, thereby mitigating the substantial memory overhead typically associated with backpropagation. For instance, inference-only methods like zeroth-order optimization can reduce memory usage by up to 12.5 times compared to BP-based methods [187]. Despite the promise shown by BP-free training methods in optimizing LLMs, they are still in the early stages of development. A significant challenge lies in the scalability of these methods to high-dimensional models, as they exhibit greater sensitivity to dimensionality and reduced robustness compared to BP-based methods [188]. Various optimizations have been proposed to address these challenges, including tuning the low intrinsic dimension of LLMs [189]. The potential impacts of backpropagation-free methods are significant. Firstly, they can drastically reduce the computational cost and memory requirements associated with training large-scale models. This

TABLE III
TAXONOMY OF PERSONALIZED AND EFFICIENT FEDERATED LLM RESEARCH

Reference	Heterogeneity	Knowledge Transfer Techniques	Model	Main Dataset	Key Contribution
Fed-PepTAO [167]	Data heterogeneity	Prompt tuning	GPT2, LLaMA 7B	10 commonly-used tasks	Propose a parameter-efficient prompt tuning approach with Adaptive Optimization,to enable efficient and effective FL of LLMs.
Profit [168]	Data heterogeneity	Prompt tuning	PaLM	(HHF, MHF, LHF) SNI	Benchmarking fundamental FL algorithms (FedAvg and FedSGD) plus personalization using LLM PEFT methods under varying levels of data heterogeneity.
FedDAT [169]	Data heterogeneity	Prompt-tuning with KD	ViLT	14 Vision-Language benchmarks	Finetuning framework tailored to heterogeneous multi-modal FL leveraging a Dual-Adapter Teacher (DAT) to address data heterogeneity
FedRA [170]	System heterogeneity	Adapter with allocation matrix	ViT	NICO++, DomainNet	Propose a novel federated tuning algorithm, FedRA, to meet the needs of heterogeneous clients with varying computation and communication resources.
pFedLoRA [171]	Model heterogeneity	LoRA	Heterogeneous CNNs	CIFAR-10, CIFAR-100	Propose a novel and efficient model-heterogeneous personalized FL framework based on LoRA tuning using homogeneous small local adapters.
pFedPG [172]	Data/system heterogeneity	Prompt tuning	ViT-B/16	Office-Caltech10, DomainNet	Jointly optimizes the stages of personalized prompt adaptation locally and personalized prompt generation globally.
Heterogeneous-LoRA [173]	System heterogeneity	LoRA	PaLM2	MSC	Examine federated fine-tuning with homogeneous LoRA ranks, and deploy hetergeneous ranks across clients, which is simple yet effective.
FedBERT [175]	-	Split learning	BERT, GPT2	GLUE	Grant clients with limited computing capability to participate in pre-training a large model combining FL and split learning.
FedBFPT [176]	System heterogeneity	Compression	BERT	S2ORC, ERC, Rct-20k	Effcient FL framework for further pre-training the BERT language model on the client without sharing private corpora
FedOBD [177]	System heterogeneity	Compression	Transformer based	CIFAR, IMDB	It decomposes large-scale models into semantic blocks so that FL participants can opportunistically upload quantized blocks

makes it feasible to train and fine-tune models on resource-constrained devices. Secondly, these methods can enhance the robustness and generalization capabilities of models by introducing diverse optimization strategies that are less prone to overfitting.

FwdLLM [190] is a pioneering study that integrates backpropagation-free (BP-free) training, specifically zeroth-order optimization, with methods that are efficient in terms of the amount of updated parameters. This combination is essential for scaling up to the era of LLMs. The BP-free approach is particularly compatible with PEFT techniques, which require only a minimal number of parameters to be fine-tuned. Furthermore, FwdLLM is designed to distribute computational tasks across devices in a systematic and adaptive manner, striking an optimal balance between the speed of convergence and the accuracy of the model. Consequently, FwdLLM facilitates federated training of LLMs with billions of parameters on standard mobile devices.

ZOOPFL [191] aims to investigate the impact of LLMs on FL performance and efficiency. The researchers conducted a series of experiments on four typical NLP tasks using different LLMs and FL methods. It was found that LLMs can significantly improve the accuracy and generalizability of FL models, but also introduce high computational cost and communication overhead. The results suggest that LLMs should be carefully selected and adapted for FL scenarios, and that novel techniques such as compression, quantization, and distillation should be applied to reduce the resource consumption.

FEDBPT [192] aims to address the challenges of applying FL to fine-tune LLMs, such as restricted model parameter access, high computational requirements, and communication overheads. The framework does not require clients to access the model parameters. Rather, it trains optimal prompts using gradient-free optimization methods, which reduces the number of variables to be communicated, enhances communication efficiency, and minimizes computational and storage costs.

FedKSeed [193] is a zeroth-order optimization-based FL method for LLM, which enables full-parameter tuning of billion-parameter LLMs on federated devices with extremely low communication cost. It communicates only K seeds and their corresponding scalar gradients between the server and the clients. Moreover, it investigates the differentiated importance of perturbations in ZOO, and proposes a simple and effective strategy that samples seeds with non-uniform probabilities, which reduces the number of required seeds.

F. Limitations and Lessons Learned

The preceding section has provided a comprehensive overview of the state-of-the-art in LLM pre-training and fine-tuning with FL. In this subsection, we discuss the limitations of current methodologies and the lessons learned from recent advancements in this field.

The diverse range of frameworks examined underscores the ongoing need for innovation in FL systems. Each framework provides distinct solutions to the challenges of privacy, efficiency, and scalability. The evolution of frameworks like FedLLM, FATE-LLM, and FS-LLM highlights the critical importance of infrastructure development that not only supports but also enhances the capabilities of LLMs within federated environments. Currently, most research is centered

around fine-tuning LLMs using FL or combining it with PEFT to achieve personalized FL. However, studies on pre-training LLMs with FL remain scarce due to the significant computational and communication costs involved. In an era of diminishing data availability, federated LLM pre-training presents a promising and practical approach, as it allows the incorporation of private domain data while safeguarding data privacy.

Furthermore, the establishment of benchmarks such as FedIT and the utilization of evaluation metrics in OpenFedLLM underscore the necessity for standardized testing environments. These benchmarks are crucial for accurately assessing the performance of federated LLM systems and ensuring that advancements are both meaningful and measurable. In the future, there will be a need for larger-scale and more standardized benchmarks to further enhance the evaluation process.

Additionally, to reduce the resource overhead of federated LLMs, PEFT and back-propagation-free methods have been explored. PEFT-based approaches, such as FedPETuning and FedTune, offer a promising path to reducing computational costs and enabling training on devices with limited resources, thereby broadening the accessibility of LLMs. The exploration of back-propagation-free methods like FwdLLM and FedKSeed introduces a paradigm shift in the training of LLMs. These methods can be combined with existing techniques such as model quantization, pruning, and knowledge distillation to further reduce resource overhead. Additionally, the heterogeneity present in FL environments poses a significant challenge. Solutions like Fed-PepTAO, Profit, and FedDAT, which integrate with LLM PEFT methods, showcase the potential of personalized approaches to address the heterogeneity in data, systems, and models.

V. Federated LLM Privacy and Robustness: State-of-the-Art

One of the crucial aspects in the research of FL and LLM is how to ensure both privacy and robustness. FL aims to address the privacy issues in ML training, and both FL and LLM inherently face their distinct challenges to privacy and robustness. The interaction between LLMs and FL systems may exacerbate the potential vulnerabilities in these systems, resulting in new challenges. In this section, we will first present the privacy leakage and security problems that Federated LLM may encounter, and then we will summarize the defense methods against these issues.

A. Federated LLM Privacy

FL is a training approach for models that prioritizes privacy, eliminating the need for data exchange and allowing members to freely join or leave the network. Nonetheless, recent inquiries indicate that FL might not be completely reliable in protecting privacy. Analyzing from the FL standpoint, the current protocols of FL exhibit susceptibilities in two distinct areas. Initially, a hostile server could aim to retrieve confidential data from individual contributions incrementally, sway the training operations, or alter the collective understanding of the global model weights. Secondly, a participant with adversarial intentions might infer confidential data about other members, and disrupt the aggregation process of the global model weights. Specifically, the act of sharing gradients during training can inadvertently reveal private data, potentially resulting in significant privacy leakage [194], [195], which may affect not just external entities but also the central server managing the process [196], [197]. It has been noted that even a minimal subset of gradients can unveil extensive details about the local dataset [198]. Furthermore, recent works also have shown that an adversary could, through gradient observations alone, reconstruct the original training dataset [195], [199]. On the other hand, LLMs like GPT-3 carry potential privacy risks due to their design to assimilate and generate text from extensive, varied datasets. These models might inadvertently encode and disclose confidential details found within their training data, leading to privacy issues in text creation process. Challenges including unintended data memorization and information leakage are critical [200]. Thus, it is crucial to strike a tradeoff between the inference performance of these advanced LMs and the capabilities to protect user privacy, to ensure their reliable and ethical deployment across different sectors. We propose a basic taxonomy that facilitates the comprehension of the various types of privacy attacks, categorized by the attacker's objectives.

1) Training Data Recovery Attacks: Training data recovery attacks, also referred to as reconstruction attacks, target the retrieval of a client's LLM training data within a practical FL environment. These attacks are predominantly gradient-based, exploiting the data transmitted between clients and the federated server. Deep learning typically utilizes optimization algorithms reliant on gradients, and federated participating clients transmit their gradients to the federated server each round, adhering to a federated Stochastic Gradient Descent (SGD)-based training protocol. Attackers with access to these gradients, or the ability to deduce gradient information, may be able to reconstruct the confidential training data. It has been demonstrated by several studies [201], [202], [203], [204] that gradients from deep learning models can be exploited to reconstruct original private training data within an FL framework. These techniques are primarily effective with image data, and there is limited research on gradient leakage for LLMs, particularly in a federated context. TAG [205] is designed to address and resolve the gradient attack issue on Transformer-based LMs, aiming to recover local training data. TAG introduces a quantitative evaluation approach for the NLP gradient attack challenge, utilizing metrics such as Recovery Rate, ROUGE-1, ROUGE-2, ROUGE-L, and runtime to measure the attack algorithm's success. According to these metrics, TAG has achieved a Recovery Rate that is 1.5 times higher and a ROUGE-2 score that is 2.5 times greater than previous methods. Tests conducted on models like Transformer, TinyBERT_4, TinyBERT_6, BERT_{BASE}, and BERT_{LARGE} using the GLUE benchmark have confirmed TAG's effectiveness.

LAMP [206] leverages language model priors to retrieve private text from gradients. The fundamental concept behind this type of attack is to integrate the predictive capabilities of a

language model with a search strategy that oscillates between continuous and discrete optimization phases. In particular, it produces a list of candidate sentences by applying different transformations on the token sequence (e.g., moving a token) and selects a candidate that minimizes the joint reconstruction loss and perplexity, which reflects the likelihood of the text in a natural distribution. The experiments are based on BERT_{LARGE} and GPT-2 and the experiment results illustrate the effectiveness of this method in extracting text from state-of-the-art transformer models on several common datasets, achieving up to 5 times more bigrams than previous work.

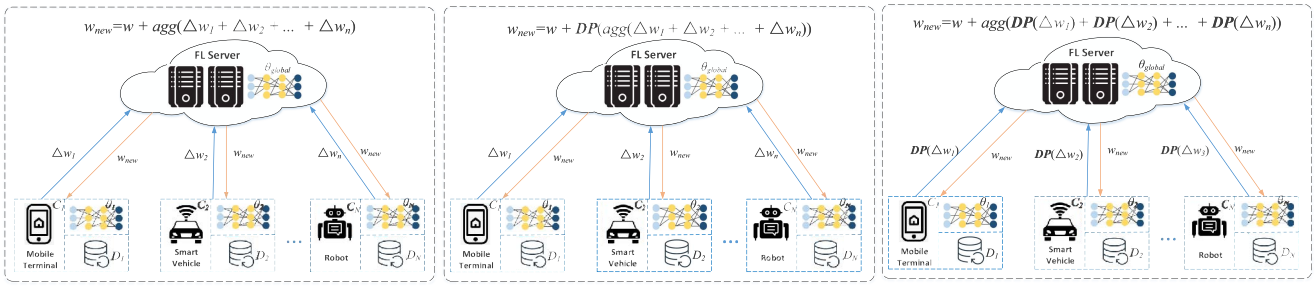
FILM [207] is the first method to demonstrate the possibility of recovering text from large batch sizes of up to 128 sentences. Unlike image-recovery methods that are designed to match gradients, it adopts a different approach that first extracts a set of words from gradients and then directly reconstructs sentences based on beam search and a prior-based reordering strategy. Three defense methods: gradient pruning, DPSGD, and a simple approach to freeze word embeddings are evaluated. Both gradient pruning and DPSGD result in a significant loss of utility. However, when fine-tuning a public pre-trained LM on private text without updating word embeddings, it can successfully defend the attack with minimal data utility loss.

The DECEPTICONS [208] framework introduces a novel attack strategy that compromises user privacy by transmitting harmful parameter vectors. This method is effective even with mini-batches, multiple users, and extended sequences. It uniquely leverages the Transformer architecture and token embeddings to separately recover tokens and positional embeddings, resulting in high-quality text reconstruction. This approach underscores the significance of the malicious server threat model, emphasizing the vulnerability of text applications using Transformer models to privacy breaches. The experiments demonstrate the feasibility of recovering all tokens and most of their absolute positions, even in large sequences and with models that are only 10% the size of BERT.

FLAT-CHAT [209] is a novel and efficient gradient flattening attack method. It is inspired by the sparsity property of gradients from the last linear layer, and applies a matrix flattening operation on the gradient matrix. The method is based on a theory that the flattened gradient vector elements follow a two-cluster Gaussian Mixture Model and three observations on the statistical properties of the distribution. To reduce the risk, two defense methods are evaluated, gradient freezing [207] and Differentially Private Stochastic Gradient Descent [210], against the attack. The former method is a robust defense method but it compromises the models' performance. The latter can mitigate our attacks while achieving improved model performance.

2) Membership Inference Attacks: Membership Inference Attacks (MIAs) are designed to ascertain whether a particular dataset was utilized during the training of a model, based on the client's model and some data. In an FL framework, both active and passive MIAs can be conducted [194], [211]. Passive MIAs entail monitoring the model's updated parameters and deducing information without interrupting the learning process. On the other hand, active MIAs involve tampering with the FL training protocol, constituting a more aggressive form of assault on the other participants. Regarding LMs, MIAs are primarily focused on text generation and subsequent text classification tasks [212], [213]. While LMs are generally resistant to basic probing techniques [214], they are still vulnerable to privacy threats from MIAs specifically crafted for LMs. A common technique in Membership Inference Attacks (MIA) is known as the threshold attack. This method is particularly relevant for word embedding models, which are susceptible to privacy breaches [215], [216]. It works by transforming text data into vector embeddings and then calculating a similarity score between these vector pairs. If the average similarity score exceeds a predefined threshold, the data is considered to have been part of the training set. The study by Song and Raghunathan [217] examined the vulnerability of three prominent word embedding models, namely Word2Vec [218], FastText [219], and GloVe [220], and all of which were trained using the Wikipedia corpus by Mahoney [221]. Additionally, they scrutinized a dual-encoder framework for sentence embeddings [222], which was trained on the BookCorpus dataset. The findings highlighted that these models, often considered merely as beneficial tools for model training, could also pose risks of privacy breaches. Other works [223], [224], [225] also employ a similar idea to use some form of reference model to compute the threshold test statistic. Another one of the methods for MIA relies on the shadow model technique, which typically builds several "shadow" models that emulate the target model, given a known training dataset and its membership labels. The attack model is trained using labeled data, distinguishing between member' (part of the training set) and non-member' (not part of the training set), along with the inputs and outputs from shadow models. Song and Shmatikov [226] pioneered the study of membership inference in natural language text generation, followed by Meeus et al. [227], Carlini et al. [228], and Abascal et al. [229]. Moreover, several defense mechanisms against MIAs, including information perturbation, have been suggested to shield natural language models at various phases of the target model's development. However, the these attack models mainly concentrated on LMs and only a few on LLMs, and there is hardly any work that discusses the integration of FL and LLMs. It is imperative to conduct comprehensive research and scrutiny to ascertain the effects of the attack methods on federated LLMs and to identify viable countermeasures.

3) Property Inference Attacks: Property inference attacks are another potential privacy threat for LLMs, although they have received less attention than membership inference and training data extraction attacks. Attribute inference attacks in the context of FL are designed to deduce specific characteristics of a client or the collective attributes of participants that are not directly related to the primary function of the machine learning model. The objective is to uncover personal or demographic details that are intended to remain confidential. For instance, these attacks might aim to infer sensitive information such as an individual's name, contact number, residential address, or private financial and medical records. Therefore, attribute leakage has also been a serious problem



(a) FL Without DP (b) FL with GDP (c) FL with LDP

Fig. 7. FL without DP and with different DP mechanisms.

to be solved in LLMs. Staab et al. [230] conducted a comprehensive research on the attack risks. It leverages publicly available content authored by individuals, like messages on digital social platforms. This information is incorporated into a structured prompt that instructs an LLM to deduce the personal characteristics of the post's author. Utilizing the data from user profiles, which include details such as age, educational background, gender, profession, and geographical location, GPT-4 was able to correctly identify these attributes with a Top-1 accuracy rate of 84.6%.

B. Defence Methods Against Federated LLM Privacy Attacks

Privacy preservation methods have been thoroughly investigated in the machine learning field, but it becomes even more complex in FL settings, where factors such as intermittent power and network availability, and heterogeneous data distributions, affect the learning process. In this section, we review some of the mainstream techniques for preserving privacy, such as differential privacy (DP), homomorphic encryption (HE), and secure multi-party computation (SMPC), and how they can be integrated in federated LLM scenarios.

1) Differential Privacy: DP is a technique that was initially developed for the single database setting, where a database server responds to each query with a randomized answer that preserves privacy [231]. Unlike encryption-based methods, DP achieves a balance between privacy-preserving and model accuracy by adding noise to the data in a manner that prevents an adversary from reconstructing the original data and maintains a high level of utility. It ensures that any output from a differentially private algorithm is nearly the same, whether or not an individual's data is included in the dataset. Formally, a randomized algorithm $M : D \rightarrow R$ satisfies ϵ-differential privacy if for any two adjacent datasets $x, y \in D$ that differ by only one record, and any subset of outputs $S \subseteq R$, the following inequality holds:

$$\Pr[M(x) \in S] \leq e^{\epsilon} \Pr[M(y) \in S] \tag{5}$$

This indicates that the likelihood of a specific result being produced by the algorithm is limited by a factor of e^{ϵ}, which is independent of whether any single entry is included in the dataset or not. The parameter ϵ controls the degree of privacy: lower values of ϵ correspond to more robust privacy safeguards, albeit with the trade-off of increased distortion in the final result.

There are mainly two types of DP in FL scenarios, namely Global Differential Privacy (GDP) and Local Differential Privacy (LDP). GDP [232], [233], [234], [235], [236] has an advantage of preserving privacy with a limited cost to model performance, as it adds limited noise to the aggregated data, ensuring a good statistical distribution. On the other hand, LDP [237], [238], [239], [240], [241] methods offer a stronger privacy guarantee than GDP-based FL methods, as individuals can apply noise to their sensitive data locally to meet DP standards prior to sharing it with a potentially untrustworthy data collector. Moreover, a variety of LM works adopt DP to protect privacy against training data recovery attacks, membership inference attacks, and property inference attacks, which include data perturbation and output perturbation [242]. By using the stochastic gradient descent optimization algorithm [243], models with DP can reduce the empirical privacy leakage while maintaining comparable model utility in the non-DP setting. Fig. 7 shows the different DP mechanisms. In addition, a few studies have also explored the integration of LLM and FL using DP for preserving privacy.

Basu et al. [244] investigate the impact of applying DP in FL scenarios on training contextualized language models (BERT, ALBERT, RoBERTa and DistilBERT). They benchmark the effect of privacy mechanisms such as DP on the performance of the federated BERT-based models. The experiments with different privacy budgets show how the privacy budgets influence the utility of models trained on Tweets related to depression and sexual harassment. The authors provide guidance on how to train NLP models privately and what architectures and setups yield more favorable privacy-utility trade-offs.

Basu et al. [245] propose a financial text classification system that preserves privacy, using transformers (BERT and RoBERTa) with differential privacy, in both centralized and FL scenarios, testing different privacy budgets to examine the privacy-utility trade-off and their performance in classifying financial document-based text sequences. For the federated scenarios, both IID and non-IID data distributions are explored.

DP-LoRA [246] is a novel FL algorithm designed for LLMs. It employs a Gaussian mechanism to add noise in weight updates, which preserves individual data privacy and enables collaborative model training. Furthermore, DP-LoRA reduces communication costs by using low-rank adaptation,

which minimizes the amount of updated weights transmitted during distributed training. The experiments on various LLMs across medical, financial, and general datasets show that DP-LoRA can effectively satisfy strict privacy requirements while reducing communication overhead. The method ensures data privacy in LLM fine-tuning through feasible FL approaches, which allow multiple parties to securely improve LLMs.

2) Homomorphic Encryption: Homomorphic encryption (HE) allows for computations to be performed on encrypted data, maintaining the homomorphic trait, which means the outcome, once decrypted, matches the result of operations conducted on the original, unencrypted data. The homomorphic properties are mathematically defined as follows:

$$E_{pk}(m_1 + m_2) = c_1 \oplus c_2, \tag{6}$$

$$E_{pk}(a \cdot m_2) = a \otimes c_2, \tag{7}$$

where a is a constant, m_1, m_2 denote the original messages that require encryption, and c_1, c_2 refer to the resultant encrypted messages corresponding to m_1 and m_2, respectively. Homomorphic encryption can be categorized into partial homomorphic encryption (PHE), somewhat homomorphic encryption (SHE), and fully homomorphic encryption (FHE) based on the type and number of ciphertext operations that they support. PHE only supports one kind of ciphertext homomorphic operation, such as additive homomorphic encryption (AHE) or multiplicative homomorphic encryption (MHE), exemplified by Paillier [247] and ElGamal [248], respectively. SHE extends this capability to support an unlimited number of additions and at least one multiplication in the encrypted domain, and it can evolve into an FHE system through bootstrapping. FHE, which adheres to Gentry's framework, is capable of executing an indefinite sequence of both additions and multiplications on ciphertexts. HE is extensively utilized, especially for enhancing the security of learning processes by allowing computations on encrypted data, thus safeguarding against privacy breaches in FL involving LLMs. Nonetheless, the computational operations on encrypted data introduce significant overheads in terms of memory usage and processing time, necessitating a balance between security and performance in HE-based systems.

3) Secure Multi-Party Computation: The concept of Secure Multi-Party Computation (SMPC) emerged from the millionaire's dilemma as outlined in [249]. SMPC's objective is to facilitate collaborative computation of a function among multiple data proprietors who lack mutual trust, all while maintaining the confidentiality of their respective datasets. The foundational mechanisms that facilitate the SMPC framework include Garbled Circuit (GC) [250], Oblivious Transfer (OT) [251], and Secret Sharing (SS) [252]. However, these techniques have drawbacks and often need to be combined with other techniques to build efficient SFL algorithms. Generally, SMC methods are known for their high privacy and accuracy levels. However, they incur high computation and communication costs, which may discourage participation. Another major challenge for SMPC-based schemes is the need for all participants to coordinate synchronously throughout the LLM training process. This multiparty interaction model may not be suitable for practical scenarios, especially under the typical participant-server architecture in FL settings. Moreover, SMC-based protocols can enable a group of participants to jointly perform calculations on a shared function without disclosing individual inputs, except for inferences made from the output [253]. However, SMPC is not entirely foolproof against data leakage, which calls for the integration of DP mechanisms into the collective protocol to mitigate such vulnerabilities [254], [255]. Despite these challenges, SMPC remains a promising strategy for protecting privacy in FL involving LLMs.

C. Federated LLM Robustness

Robustness is the ability of the LLM to produce the desired content accurately, even under various types of attacks. Unlike privacy attacks that aim at data confidentiality, these attacks on robustness are not interested in data access, but in manipulating the model's output to mislead users and achieve the attackers' malicious goals. In this section, we discuss the robustness issues faced by federated LLMs under three typical attack methods, namely Byzantine attacks, poisoning attacks, and prompt attacks.

1) Byzantine Attacks and Defences: Adversarial attacks for robustness can be classified into two main categories, depending on the attacker's objective, namely untargeted attacks [256], [257], [258] and targeted attacks [112], [259], [260], [261], [262]. Byzantine attacks [263], [264], [265], [266] are usually defined as untargeted attacks that send maliciously crafted gradients to the model aggregator, aiming to degrade the performance of the global model or compromise its integrity. In this scenario, the server cannot verify the trustworthiness of the clients. Byzantine problems often arise during the client update phase. Certain clients might be vulnerable to external attacks or internal errors. Such compromised clients are capable of sending tainted updates to the server. If these malicious updates are inadvertently merged by the server, it could derail the entire federated optimization workflow. The concept of a Byzantine attack is formally described as follows:

$$\Delta w_i = \begin{cases} * & \text{if } i\text{th participant is Byzantine} \\ \nabla F_i(w_i) & \text{otherwise,} \end{cases} \tag{8}$$

where "*" represents any values, Δw_i denotes the gradient update, and F_i represents the local model objective function of participant i. The impact of the Byzantine attacks on distributed learning can be described as follows:

$$w = w - \Lambda\big(\Delta w_1, \Delta w_2, \ldots, \Delta w_p\big). \tag{9}$$

Byzantine Detection and Robust Aggregation are two prevalent strategies to counteract Byzantine attacks within FL. The primary goal of robust aggregation approaches [267], [268], [269] is to minimize the influence of Byzantine clients on the collective model update process. These methods presuppose that the corrupted updates are geometrically distant from the legitimate ones. Consequently, the focus is on developing an aggregation rule robust enough to mitigate the impact of these attacks. For instance, in distributed learning environments, algorithms like Krum [270]

and Bulyan [271] select local updates closest in Euclidean distance to the majority and use them for the global model update. Nonetheless, these robust aggregation methods often experience a decline in performance when a significant number of clients are compromised or when the client data is highly Non-IID. This necessitates further exploration and enhancement of robust aggregation techniques. In contrast, Byzantine detection methods are designed to pinpoint and eliminate harmful local updates, thereby preventing compromised clients from impairing the FL system [272], [273], [274]. These detection-based methods tend to offer greater resilience than their aggregation-focused counterparts. However, these approaches require extra computational resources and data demand on the server and client side. As a common problem in FL security related work, it is rarely discussed in more complex and unpredictable federated LLM scenarios, which makes it an open problem that calls for further studies on federated LLM robustness.

2) Poisoning Attacks and Defences: Poisoning attacks represent a common form of targeted disruption in FL. These attacks are twofold: data poisoning, which occurs during the collection of local data, and model poisoning, which takes place throughout the local model training phase. Specifically, model poisoning encompasses data poisoning within FL environments, as it modifies a portion of the updates transmitted to the model during any iteration. Conventionally, poisoning attacks on mainstream ML models are designed to deceive the model by tampering with the training data, often targeting classification models. For instance, attackers might contaminate a spam filter by incorporating "good" words into the training dataset [275], [276], or they might compromise network intrusion detection systems [277]. Recent studies have revealed that LLMs are particularly susceptible to poisoning, largely because their training data is predominantly sourced from the Interneta platform where content can be freely posted, thus exposing it to potential poisoning. Research has demonstrated the feasibility of poisoning expansive datasets such as LAION-400M [278], COYO-700M [279], and Wikipedia by domain purchases or crowdsourcing efforts [280]. It has been shown that contaminating a mere 0.1% of unlabeled data in semi-supervised learning can cause the model to incorrectly classify any given example during testing [281]. Moreover, even a slight 0.01% dataset contamination can cause models like CLIP to misclassify test images [281]. Backdoor attacks, a subset or variation of poisoning attacks, threaten the integrity of a model by embedding harmful functions within it using poisoned samples. These attacks can trigger inappropriate model behavior in response to specific inputs while maintaining normal function otherwise [282]. While data poisoning poses a challenge for LLMs due to the vast volume and stringent management of training data, alternative backdoor attack methods remain a viable threat. These methods introduce malicious logic into the model by altering inputs during testing, potentially leading to targeted misclassification when LLMs execute certain tasks [283], [284].

A few studies have pioneered the exploration of poisoning attacks in the scenario of federated LLM. FedMLSecurity [285] is an FL security module of FedML, which consists of two main components: FedMLAttacker and FedMLDefender. It allows for the evaluation of various attack methods in FL, such as Byzantine attacks, and label flipping backdoor attack, and defense mechanisms such as Krum (and m-Krum) and geometric median. Furthermore, FedMLSecurity supports a broad spectrum of ML models, including basic ResNet and GAN and shows the versatility of FedMLSecurity for LLMs and real-world applications through experiments.

Li et al. [286] introduce a new backdoor attack strategy for HFL, named Fed-EBD, that eliminates the need for compromising any client or sustaining long-term involvement in the FL process. This strategy implants and disseminates the backdoor via a synthetic public dataset created by a foundation model, which could elude existing backdoor countermeasures in FL by simulating normal client behaviors. Furthermore, Li et al. [287] assess the robustness of FL integrating LLMs by measuring their susceptibility to backdoor attacks. Based on this, they devise an attack that does not demand the attacker to fully subvert any client or persistently partake in the long-term FL process. It is efficacious in realistic FL settings, as the backdoor is embedded and transmitted to each client at the FL initialization and it is difficult to discern due to the limited research on the robustness of the LLMs. Another similar work by Wu et al. [288] also specializes in novel backdoor attacks for federated LLMs.

Zhou et al. [289] propose a robust pre-training strategy for foundation models that can resist attacks without demanding downstream users to adopt additional defensive measures. The defense strategy aims to increase the feature distance between poisons and targets. This is accomplished by altering the feature distribution of the pre-trained model through two methods, namely augmenting the feature distance between samples of different classes and generating poison samples with adversarial samples to shrink the feature distance between poison samples and clean samples.

Huang et al. [290] propose a secure distributed large language model (LLM) framework based on model slicing. This framework employs the Trusted Execution Environment (TEE) on both the client and server sides, incorporating the fine-tuned structure (either LoRA or the embedding of P-tuning v2) within the TEE. Secure communication is facilitated between the TEE and general environments through lightweight encryption. To further reduce equipment costs and enhance model performance and accuracy, the authors introduce a split fine-tuning scheme. Specifically, the LLM is partitioned by layers, with the latter layers placed in a server-side TEE, thereby eliminating the need for a TEE on the client side.

All the above works attempt to analyze the vulnerability of federated LLMs and design new poison attack methods. The vulnerabilities pose new threats to the security and reliability of the federated LLMs system. However, there are few works in this direction currently. Therefore, exploring how to discover new vulnerabilities and achieve corresponding good effects of adversarial defense is worthwhile.

3) Prompt Attacks and Defences: LLMs are sensitive to the engineering of prompts, and it has been shown that LLMs can be inconsistent with their answers when prompted differently. Prompt attacks involve strategically designing and

TABLE IV
TAXONOMY OF FEDERATED LLM PRIVACY AND ROBUSTNESS RESEARCH

Reference	Classification	Core method	Model	Main Dataset	Key Contribution
Basu *et al.* [244]	Privacy	DP	BERT, RoBERTa	Depression Dataset	Study the effects that the application of DP has, in FL setup on training contextualized language models.
Basu *et al.* [245]	Privacy	DP	BERT	Financial Phrase Bank	Propose a contextualized transformer (BERT and RoBERTa) based text classification model integrated with privacy features such as Differential Privacy (DP) and FL.
DP-LoRA [246]	Privacy	DP	GPT2, BERT, ChatGLM-6B, LLAMA2-7B	SlimPajama, Medical, Finance	DP-LoRA preserves data privacy by employing a Gaussian mechanism and optimizes communication efficiency via low-rank adaptation.
FedMLSecurity [285]	Privacy/ robustness	-	BERT, Pythia-1B	20News, PubMedQA	FedMLSecurity is an end-to-end benchmark (an open-source library) designed to simulate adversarial attacks and corresponding defense mechanisms in FL.
Li *et al.* [286]	Robustness	Backdoor	DistilBERT	SST-2 AG-News	Introduce a novel backdoor attack mechanism for HFL that circumvents the need for client compromise or ongoing participation in the FL process.
Li *et al.* [287]	Robustness	Backdoor	DistilBERT	SST-2 AG-News	Investigate the robustness of FL incorporating FMs by assessing their susceptibility to backdoor attacks, and propose a novel attack in FM-FL.
Wu *et al.* [288]	Robustness	Backdoor	DistilBERT, GPT-4	AG-News	Conduct the firstd investigation on the vulnerability of FM-FL under adversarial threats, and introduce a novel attack strategy that exploits FM safety issues.
Huang *et al.* [290]	Robustness	-	ChatGLM-6B	CHIP-CTC, KUAKE-IR	This framework employs the TEE on both the client and server sides, incorporating the fine-tuned structure (either LoRA or the embedding of P-tuning v2) within the TEE.

manipulating input prompts to modify the output of LLMs. The intent behind this tactic is to direct the model towards producing specific outputs or achieving particular objectives. Even models that have undergone extensive training may yield deceptive or harmful outputs when presented with certain tailored prompts. One of the common methods of this kind of attacks is prompt injection [291], [292], [293], [294], where the attacker gains control over the output of a language model, enabling them to dictate the content it generates. This method involves bypassing safeguards by using specially crafted prompts that cause the model to ignore previous instructions or execute specific tasks. Such security loopholes could result in various adverse outcomes, including the exposure of sensitive data, unauthorized system entry, or other forms of security problems. For instance, [295] has demonstrated that a GPT model's responses can be swayed by introducing specially engineered adversarial disturbances, affecting its text classification capabilities. In [296], the model might be configured to avoid performing certain sensitive tasks, like modifying a user's password. However, prompt injection attack using certain prompts, e.g., instructing the LLM to "ignore previously established restrictions," the assistant could be manipulated into carrying out these prohibited actions.

For such prompt-based attack methods in LLMs, limited studies explored the corresponding defense strategies. Some preventive measures [297] are suggested, such as preprocessing the data prompt to remove the injected task's instruction/data, and/or redesigning the instruction prompt itself. To counter adversarial prompts, several techniques can be used, for instance, paraphrasing [298], re-tokenization [299], data prompt isolation [299], and instructional prevention [299]. Prompt attacks may affect the fine-tuning and inference applications of federated LLMs, but they are mainly due to the security issues of LLMs, and they barely involve the FL process. Defending from the perspective of LLMs alone can prevent such attacks, so this topic will not be discussed further in this paper.

D. Limitations and Lessons Learned

The exploration of federated LLMs is a dynamic field that presents unique challenges and opportunities. This section aims to distill the lessons learned from current research and practice regarding the privacy and robustness of federated LLMs, highlighting areas that require further attention and innovation.

The integration of FL and LLMs necessitates adaptive defenses [300] due to the dynamic nature of cyber-attacks. Federated LLMs, which involve distributed training across multiple devices, are particularly vulnerable to novel attack vectors. Static security measures are inadequate; instead, adaptive defenses that evolve in response to emerging threats are essential. These defenses can leverage real-time data from various nodes to detect and mitigate attacks, ensuring the robustness and security of federated LLMs. Additionally, developing secure and efficient federated LLMs requires a multidisciplinary approach. For instance, combining insights from cryptography, machine learning, and network security is crucial. Cryptographic methods can secure data during transmission and storage, while machine learning techniques enhance model performance and resilience. Network security ensures the integrity of data exchanges between nodes. By integrating these disciplines, researchers can address the complex challenges of federated LLMs, creating robust and secure systems.

As federated LLMs become more complex, transparency and explainability are paramount. Understanding the decision-making processes of these models helps identify vulnerabilities and build trust. Explainable AI (XAI) techniques [301], [302]

can be applied to federated LLMs to interpret model outputs and provide insights into their functioning. This transparency is crucial for debugging, improving model performance, and ensuring that federated LLMs operate as intended. It also fosters trust among users and stakeholders by elucidating how decisions are made.

Moreover, balancing privacy and utility is a central challenge in federated LLMs. Privacy-preserving techniques, such as differential privacy and secure multi-party computation, are essential to protect sensitive data. However, these methods can impact model performance. Research must continue to explore innovative approaches that enhance privacy without significantly compromising utility. Achieving this balance ensures that federated LLMs remain both effective and secure, providing high-quality results while safeguarding user data.

VI. Future Research Directions

As discussed in the previous sections, integrating LLMs with FL is a novel technique that can be regarded as an emerging research area. After a thorough review of the existing works on the federated LLM training and fine-tuning process, as well as privacy and robustness mechanisms, this section explores the several key challenges, and also discusses the possible research directions to address these challenges.

1) *Efficiency of Federated LLM*: As discussed in Section II, LLMs usually have a huge amount of parameters, which poses significant challenges to their training and deployment on resource-limited clients. In order to tackle this obstacle, we have presented the state-of-the-art federated LLM approaches that leverage PEFT techniques, such as Adapter and LoRA, to fine-tune the LLMs efficiently, and the backward propagation-free methods to reduce computational costs and enhance system performance. However, several other techniques can also be leveraged with them to further improve the efficiency of Federated LLMs: (i) From the perspective of model structure, more resource-efficient model structures can be combined, such as more efficient Attention module designs [303], [304], Dynamic Neural Networks [305], [306] (e.g., Mixture-of-Experts (MoE) structure [307]) to reduce computational and memory overheads. (ii) Model compression strategies can also be applied, such as pruning [308], [309], quantization [310], [311] and knowledge distillation [181], [182] methods for LLMs, which can effectively reduce the model size without significant performance degradation. (iii) From the perspective of infrastructure, more efficient computing and inference hardware and software designs, such as parallel computing [312], KV cache utilization [313], and novel edge computing hardware, can also be used to meet the computational demands of federated LLMs by enhancing efficiency. Nevertheless, these studies mainly consider the isolated LLM scenario, combining these methods with FL and their efficacy have not yet been fully examined, which will inevitably introduce new optimization methods and new challenges.
2) *Heterogeneity of Federated LLM*: In practical large-scale Federated scenarios, there may be significant differences among data distributions, model structures, communication networks, and system edge clients, which make it difficult to achieve federated collaboration. Such heterogeneity can be classified into four categories, namely data heterogeneity, communication heterogeneity, device heterogeneity, and model heterogeneity. The research that we have reviewed in this paper mainly focuses on device heterogeneity, where most works adopt PEFT methods to adapt to different computational capabilities of client devices. However, few works have considered the impact of other types of heterogeneity. For example, various studies [101], [102], [314] indicate that the local optimization objectives of clients are not aligned with the global optimization objective due to the variations in the local data distribution. Therefore, data heterogeneity may cause local models to converge to different directions, reaching local optima instead of global optima, thus impairing the FL performance. In the scenario of federated LLM, LLMs have access to a large amount of data for training, and data heterogeneity may have a greater impact on the training and fine-tuning processes. On the other hand, data diversity could also potentially improve the model's generalization performance. Therefore, it is crucial to investigate the impact of data heterogeneity and other types of heterogeneity on federated LLM more deeply, as this type of work is still scarce.
3) *Privacy of Federated LLM:* A holistic approach that encompasses both impact evaluation and solution design is required for privacy protection, which is an growing research area that needs further improvements. As mentioned in Section V, some studies have started to investigate the privacy challenges of Federated LLM. DP and its variations demonstrate reliable and generalizable privacy protection capabilities, but they have limitations when it comes to handling complex tasks, and these approaches did not consider the heterogeneous resources of clients. Cryptographic protections such as SMPC and HE are primarily utilized during the inference phase of LLMs, and these approaches usually encounter high communication and computation costs. Furthermore, these methods have not been evaluated for the feasibility of applying a robust privacy-preserving algorithm, and developing a system that can be adapted to the federated LLM scenario. Further research needs to be conducted to obtain maximum privacy benefits throughout the entire lifecycle of LLMs.
4) *Robustness of Federated LLM:* There has been considerable thorough and comprehensive research on the robustness of LLM and FL against adversarial attacks. However, the work that considers FL-integrated LLM is scarce. The federated LLM system is evidently more large-scale and complex than conventional FL systems. Consequently, adversaries are likely to have more opportunities to exploit security vulnerabilities within federated LLM systems and launch malicious

attacks. Thus, a comprehensive assessment of the vulnerability of federated LLM to potential threats is essential. This evaluation should examine the impacts of malicious attacks, such as backdoors, Byzantine attacks, and possible novel attack methods. Moreover, from the FL perspective, a thorough evaluation of the existing defense mechanisms against emerging threats is also necessary. This evaluation should include the effectiveness of robust aggregation strategies and post-training detection methods in combating these new threats. As we look ahead to the future, sustained research and innovation in this area will be crucial to advancing the field for federated LLMs.

In addition to the potential future directions that aim to address the current challenges mentioned above, given that the research on federated LLM is still in its early stages, there are various other opportunities that are worth exploring, for example:

1) *Multimodal Models Integration:* In the rapidly evolving field of AI, LLM is one of the most popular topics. For example, millions of new GPTs in the GPT Store are tailored for specific tasks or interests, such as providing personalized trail recommendations, coding tutorials, or even generating haikus. The GPT Store facilitates the discovery and use of these custom GPTs. Besides, other multimodal models, such as GPT-4o [315], which can process and generate text, audio, and images simultaneously, Vision Transformers Models (ViTs) [316] for various downstream vision tasks, Latent Diffusion Models for high-quality images with arbitrary text-based prompts generation, and CLIP and ImageBind [317] which map different modal data into the same latent space, are also developing quickly. These foundation models are similar to LLM in that they have a large amount of model parameters and require a lot of data for training, as well as a lot of computational resources for training and deployment. This makes them compatible with FL for similar reasons. Therefore, similar heterogeneity, privacy and robustness issues also exist when integrating FL with these foundation models. However, due to the different characteristics of modal information, these issues manifest and are addressed differently. For example, different modal information may have different data distributions, dimensions, formats, and quality, which affect the training efficiency and effectiveness when adopting FL for training. Therefore, appropriate data alignment, data augmentation, and data selection strategies are needed to reduce the negative impact of heterogeneity. Building on the research findings of LLM in this paper, how to solve these problems in a broaderly ranged foundation models remains an open challenge and a potential and promising research direction.
2) *Federated Domain-specific AI agents:* Domain-specific AI agents [318], which are closely related to large LLMs, are garnering significant attention. LLMs empower intelligent agents to autonomously address complex problems. Domain-specific LLM agents, which are LLM-based agents deeply integrated with domain-specific data, provide specialized assistance in fields such as healthcare and finance. However, the data in these domains is highly sensitive and subject to stringent privacy regulations. This data is often distributed across various locations, complicating centralized training efforts. FL offers a promising solution by enabling the training and fine-tuning of models without the need to centralize data, thereby preserving privacy. Consequently, the integration of domain AI agents with FL represents a highly promising future direction.
3) *Continual Learning for Federated LLM:* LLMs need to constantly update their domain knowledge as they are applied in dynamic real world scenarios, where data distributions may change over time and cause domain and concept drift. A potential solution is to combine federated LLMs with continual learning methods [319]. Continual learning is a branch of machine learning that focuses on how to enable machine learning models to learn from new data continuously, by retaining knowledge from previous learning experiences without catastrophic forgetting them. It tackles the challenges of incrementally training a model using real-time collected data, which may vary over time and cause data drift. LLMs have the potential to achieve better generalization and representation learning, which makes them suitable for adapting to new distributions through continual learning. Future research can explore how to address the performance challenges of continual learning in federated LLM settings.
4) *Legal, Responsible, and Profitable Usage:* Federated LLMs involve multiple entities that contribute their own data, devices, and computational resources to the training process. These models are used for various content creation applications by the users. However, this rapid growth has also raised conflicts, especially regarding intellectual property (IP) rights. While some technical methods, such as watermarking for LLMs [320], [321], have been proposed, tackling these challenges still demands a multidisciplinary approach, which incorporates not only advances in machine learning and statistics, but also insights from fields such as law, ethics, and social sciences. Ensuring the lawful and ethical utilization of these trained models is a critical future direction. Furthermore, the development of a sustainable business model for federated LLMs is imperative. Although federated LLMs have numerous potential applications, such as in healthcare where patient data privacy is paramount, robust models can be developed by training LLMs on decentralized patient records across multiple hospitals without compromising sensitive information. Another potential application is in the financial sector, where federated LLMs can analyze transaction data from different banks to detect fraudulent activities while ensuring data confidentiality. However, this model must delineate strategies to render federated LLMs profitable. However, this model must delineate strategies to render federated LLMs profitable, as the

widespread adoption of this technology hinges on its economic viability. Without a clear pathway to profitability, the potential of federated LLMs may remain unrealized, limiting their impact and utility.

5) *Data Erasure for Federated LLMs:* Although LLMs are exceptionally powerful, their reliance on vast datasets can also become a liability due to privacy concerns, accuracy limitations, copyright infringement issues, and the potential propagation of societal biases. A notable example is the lawsuit filed by the New York Times against OpenAI and Microsoft for using copyrighted content in training their GPT models, sparking a controversial debate on the application of fair use rules to LLM training and highlighting the urgent need for data erasure mechanisms [322]. To comply with the "right to be forgotten" requirements stipulated by the EU's GDPR and the US's CCPA/CPRA, it is essential to incorporate data erasure mechanisms in future LLMs. Recently, there has been a surge in relevant studies discussing this issue in the context of LLMs [322], [323], [324], [325]. However, this challenge becomes particularly pronounced in FL settings, where the concerned sensitive data may not be centrally stored. Hence, future research should focus on developing a distributed approach to machine unlearning for FL-LLMs, ensuring effective data erasure while maintaining the decentralized nature of FL.

VII. Conclusion

This paper provided a comprehensive, systematic overview of recent advances on integrating FL with LLMs. We first introduce the preliminary background of FL and LLM respectively, including their development history, basic workflow, and common architectures and algorithms. We then presented the motivation for integrating FL with LLMs from multiple perspectives, as well as the benefits they can bring to each other. We also categorized and reviewed the current works from the perspective of the whole lifespan of LLMs, from training to deployment. In addition, we also classified and reviewed the current works from the perspective of privacy and robustness. Finally, we discussed the open opportunities and future directions for federated LLM research based on the comprehensive investigation of existing works.

References

[1] M. A. Ferrag et al., "Edge learning for 6G-enabled Internet of Things: A comprehensive survey of vulnerabilities, datasets, and defenses," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2654–2713, 4th Quart., 2023.

[2] M. Xu et al., "Unleashing the power of edge-cloud generative AI in mobile networks: A survey of AIGC services," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 2, pp. 1127–1170, 2nd Quart., 2024.

[3] D. Katare, D. Perino, J. Nurmi, M. Warnier, M. Janssen, and A. Y. Ding, "A survey on approximate edge AI for energy efficient autonomous driving services," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2714–2754, 4th Quart., 2023.

[4] D. C. Nguyen et al., "Enabling AI in future wireless networks: A data life cycle perspective," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 553–595, 1st Quart., 2020.

[5] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "AI models for green communications towards 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 210–247, 1st Quart., 2021.

[6] H. Du et al., "Diffusion-based reinforcement learning for edge-enabled AI-generated content services," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8902–8918, Sep. 2024.

[7] G. Xu, Z. Hao, Y. Luo, H. Hu, J. An, and S. Mao, "DeViT: Decomposing vision transformers for collaborative inference in edge devices," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5917–5932, May 2024.

[8] Y. Cheng, Z. Zhang, and S. Wang, "RCIF: Towards robust distributed DNN collaborative inference under highly lossy IoT networks," *IEEE Internet Things J.*, vol. 11, no. 15, pp. 25939–25949, Aug. 2024.

[9] A. Radford et al., "Language models are unsupervised multitask learners." OpenAI. 2019. [Online]. Available: https://github.com/openai/gpt-2

[10] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1877–1901.

[11] J. Achiam et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.

[12] A. Chowdhery et al., "PaLM: Scaling language modeling with pathways," *J. Mach. Learn. Res.*, vol. 24, no. 240, pp. 1–113, 2023.

[13] H. Touvron et al., "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[14] H. Zhang, H. Song, S. Li, M. Zhou, and D. Song, "A survey of controllable text generation using transformer-based pre-trained language models," *ACM Comput. Surv.*, vol. 56, no. 3, pp. 1–37, 2023.

[15] H. Liu, P. Peng, T. Chen, Q. Wang, Y. Yao, and X.-S. Hua, "FECANet: Boosting few-shot semantic segmentation with feature-enhanced context-aware network," *IEEE Trans. Multimedia*, vol. 25, pp. 8580–8592, Jan. 2023.

[16] K. Valmeekam, M. Marquez, A. Olmo, S. Sreedharan, and S. Kambhampati, "PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–50.

[17] N. F. Lindemann, "Sealed knowledges: A critical approach to the usage of LLMs as search engines," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, 2023, pp. 985–986.

[18] W. Nelson, M. K. Lee, E. Choi, and V. Wang, "Designing LLM-based support for homelessness caseworkers," in *Proc. AAAI Workshop Public Sector LLMs, Algorithmic Sociotechn. Design*, 2024, pp. 1–9.

[19] Z. He et al., "Exploring human-like translation strategy with large language models," *Trans. Assoc. Comput. Linguist.*, vol. 12, pp. 229–246, Mar. 2024.

[20] S. Aycock and R. Bawden, "Topic-guided example selection for domain adaptation in LLM-based machine translation," in *Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguist., Student Res. Workshop*, 2024, pp. 175–195.

[21] J. Liu, C. S. Xia, Y. Wang, and L. Zhang, "Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 21558–21572.

[22] L. Zheng et al., "Judging LLM-as-a-judge with MT-bench and Chatbot arena," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 46595–46623.

[23] S. Roychowdhury, "Journey of hallucination-minimized generative ai solutions for financial decision makers," in *Proc. 17th ACM Int. Conf. Web Search Data Min.*, 2024, pp. 1180–1181.

[24] F. Wei et al., "Empirical study of LLM fine-tuning for text classification in legal document review," in *Proc. IEEE Int. Conf. Big Data (BigData)*, 2023, pp. 2786–2792.

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

[26] D. Yang et al., "DetFed: Dynamic resource scheduling for deterministic federated learning over time-sensitive networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5162–5178, May 2024.

[27] W. Zhang, D. Yang, C. Zhang, Q. Ye, H. Zhang, and X. Shen, "$(Com)^2$net: A novel communication and computation integrated network architecture," *IEEE Netw.*, vol. 38, no. 2, pp. 35–44, Mar. 2024, doi: 10.1109/MNET.2024.3355922.

[28] Y. Cheng, Z. Zhang, and S. Wang, "FED-SDS: Adaptive structured dynamic sparsity for federated learning under heterogeneous clients," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2024, pp. 9231–9235.

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

[30] Z. Zhang, D. Cai, Y. Zhang, M. Xu, S. Wang, and A. Zhou, "FedRDMA: Communication-efficient cross-silo federated LLM via chunked RDMA transmission," 2024, *arXiv:2403.00881*.

[31] X. Liu, T. Pang, and C. Fan, "Federated prompting and chain-of-thought reasoning for improving LLMs answering," in *Proc. Int. Conf. Knowl. Sci., Eng. Manage.*, 2023, pp. 3–11.

[32] 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 Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 3rd Quart., 2021.

[33] 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 Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1622–1658, 3rd Quart., 2021.

[34] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: Challenges and applications," *Int. J. Mach. Learn. Cybern.*, vol. 14, no. 2, pp. 513–535, 2023.

[35] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Comput. Surv.*, vol. 56, no. 3, pp. 1–44, 2023.

[36] S. Pandya et al., "Federated learning for smart cities: A comprehensive survey," *Sustain. Energy Technol. Assess.*, vol. 55, Feb. 2023, Art. no. 102987.

[37] E. T. M. Beltrán et al., "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2983–3013, 4th Quart., 2023.

[38] L. Lyu et al., "Privacy and robustness in federated learning: Attacks and defenses," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8726–8746, Jul. 2024.

[39] N. Rodríguez-Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera, and E. Martínez-Cámara, "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges," *Inf. Fusion*, vol. 90, pp. 148–173, Feb. 2023.

[40] C. Zhou et al., "A comprehensive survey on pretrained foundation models: A history from BERT to ChatGPT," 2023, *arXiv:2302.09419*.

[41] Z. Xi et al., "The rise and potential of large language model based agents: A survey," 2023, *arXiv:2309.07864*.

[42] Y. Wang et al., "Aligning large language models with human: A survey," 2023, *arXiv:2307.12966*.

[43] B. C. Das, M. H. Amini, and Y. Wu, "Security and privacy challenges of large language models: A survey," 2024, *arXiv:2402.00888*.

[44] J. Fields, K. Chovanec, and P. Madiraju, "A survey of text classification with transformers: How wide? How large? How long? How accurate? How expensive? How safe?" *IEEE Access*, vol. 12, pp. 6518–6531, 2024.

[45] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, and Q. Wang, "Software testing with large language models: Survey, landscape, and vision," *IEEE Trans. Softw. Eng.*, vol. 50, no. 4, pp. 911–936, Apr. 2024.

[46] C. Cui et al., "A survey on multimodal large language models for autonomous driving," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2024, pp. 958–979.

[47] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly," *High-Confid. Comput.*, vol. 4, no. 2, 2024, Art. no. 100211.

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

[49] C. Chen, X. Feng, J. Zhou, J. Yin, and X. Zheng, "Federated large language model: A position paper," 2023, *arXiv:2307.08925*.

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

[51] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[52] Q. Li et al., "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3347–3366, Apr. 2023.

[53] C. Feng, B. Liu, K. Yu, S. K. Goudos, and S. Wan, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3582–3592, May 2022.

[54] Y. Liu et al., "Vertical federated learning: Concepts, advances, and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3615–3634, Jul. 2024,

[55] Y. Tan, Y. Liu, G. Long, J. Jiang, Q. Lu, and C. Zhang, "Federated learning on non-IID graphs via structural knowledge sharing," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 9953–9961.

[56] R. Xu and Y. Chen, "μDFL: A secure microchained decentralized federated learning fabric atop IoT networks," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 3, pp. 2677–2688, Sep. 2022.

[57] C. Ma et al., "When federated learning meets blockchain: A new distributed learning paradigm," *IEEE Comput. Intell. Mag.*, vol. 17, no. 3, pp. 26–33, Aug. 2022.

[58] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[59] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How can we know what language models know?" *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 423–438, Jul. 2020.

[60] R. Mao, Q. Liu, K. He, W. Li, and E. Cambria, "The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection," *IEEE Trans. Affect. Comput.*, vol. 14, no. 3, pp. 1743–1753, Jul.–Sep. 2023.

[61] J. M. Patel and J. M. Patel, "Introduction to common crawl datasets," *Getting Structured Data from Internet: Running Web Crawlers/Scrapers a Big Data Production Scale*. Berkeley, CA, USA: Apress, 2020, pp. 277–324.

[62] L. Gao et al., "The pile: An 800GB dataset of diverse text for language modeling," 2020, *arXiv:2101.00027*.

[63] "Wikipedia, the free encyclopedia." 2001. [Online]. Available: https://www.wikipedia.org/

[64] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.

[65] E. Nijkamp et al., "CodeGen: An open large language model for code with multi-turn program synthesis," 2022, *arXiv:2203.13474*.

[66] "BigQuery dataset." 2024. [Online]. Available: https://cloud.google.com/bigquery

[67] Z. Du et al., "GLM: General language model pretraining with autoregressive blank infilling," 2021, *arXiv:2103.10360*.

[68] Y. Zhu et al., "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 19–27.

[69] L. Ouyang et al., "Training language models to follow instructions with human feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 27730–27744.

[70] C. Zhou et al., "LIMA: Less is more for alignment," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–15.

[71] N. Carlini et al., "Are aligned neural networks adversarially aligned?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–23.

[72] E. J. Hu et al., "LoRa: Low-rank adaptation of large language models," 2021, *arXiv:2106.09685*.

[73] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," 2021, *arXiv:2101.00190*.

[74] X. Liu et al., "P-Tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks," 2021, *arXiv:2110.07602*.

[75] X. Liu et al., "GPT understands, too," *AI Open*, vol. 5, pp. 208–215, Nov. 2024.

[76] P. Villalobos, J. Sevilla, L. Heim, T. Besiroglu, M. Hobbhahn, and A. Ho, "Will we run out of data? An analysis of the limits of scaling datasets in machine learning," 2022, *arXiv:2211.04325*.

[77] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5362–5383, Aug. 2024.

[78] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3048–3068, Jun. 2022.

[79] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, pp. 1789–1819, Jun. 2021.

[80] D. Huba et al., "Papaya: Practical, private, and scalable federated learning," in *Proc. Mach. Learn. Syst.*, vol. 4, 2022, pp. 814–832.

[81] G. Liu, X. Ma, Y. Yang, C. Wang, and J. Liu, "FedEraser: Enabling efficient client-level data removal from federated learning models," in *Proc. IEEE/ACM 29th Int. Symp. Qual. Service (IWQOS)*, 2021, pp. 1–10.

[82] D.-J. Han, M. Choi, J. Park, and J. Moon, "FedMes: Speeding up federated learning with multiple edge servers," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3870–3885, Dec. 2021.

[83] Y. He and L. Xiao, "Structured pruning for deep convolutional neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 2900–2919, May 2024.

[84] H. Cheng, M. Zhang, and J. Q. Shi, "A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10558–10578, Dec. 2024.

[85] S. Kim et al., "SqueezeLLM: Dense-and-sparse quantization," 2023, *arXiv:2306.07629*.

[86] A. Kuzmin, M. Nagel, M. Van Baalen, A. Behboodi, and T. Blankevoort, "Pruning vs quantization: Which is better?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–21.

[87] Y. Gu, L. Dong, F. Wei, and M. Huang, "MiniLLM: Knowledge distillation of large language models," in *Proc. 12th Int. Conf. Learn. Represent.*, 2024, pp. 1–23.

[88] Z. Li et al., "Curriculum temperature for knowledge distillation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 1504–1512.

[89] Q. Liu et al., "When MOE meets LLMs: Parameter efficient fine-tuning for multi-task medical applications," in *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2024, pp. 1104–1114.

[90] Y. Ge et al., "OpenAGI: When LLM meets domain experts," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–30.

[91] Z. Wang, H. Xu, J. Liu, Y. Xu, H. Huang, and Y. Zhao, "Accelerating federated learning with cluster construction and hierarchical aggregation," *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 3805–3822, Jul. 2022.

[92] J. Nguyen et al., "Federated learning with buffered asynchronous aggregation," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 3581–3607.

[93] C.-H. Hu, Z. Chen, and E. G. Larsson, "Scheduling and aggregation design for asynchronous federated learning over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 874–886, Apr. 2023.

[94] X. Ma, Q. Wang, H. Sun, R. Q. Hu, and Y. Qian, "CSMAAFL: Client scheduling and model aggregation in asynchronous federated learning," in *Proc. IEEE Int. Conf. Commun.*, 2024, pp. 274–279.

[95] G. Damaskinos, R. Guerraoui, A.-M. Kermarrec, V. Nitu, R. Patra, and F. Taiani, "Fleet: Online federated learning via staleness awareness and performance prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 5, pp. 1–30, 2022.

[96] J. Liu et al., "FedASMU: Efficient asynchronous federated learning with dynamic staleness-aware model update," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, 2024, pp. 13900–13908.

[97] M. Amadeo, C. Campolo, A. Molinaro, G. Ruggeri, and G. Singh, "Mitigating the communication straggler effect in federated learning via named data networking," *IEEE Commun. Mag.*, vol. 62, no. 11, pp. 92–98, Nov. 2024.

[98] S. Li, D. Yao, and J. Liu, "FedVS: Straggler-resilient and privacy-preserving vertical federated learning for split models," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 20296–20311.

[99] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, and Y. Zhao, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[100] X. Guo et al., "VeriFL: Communication-efficient and fast verifiable aggregation for federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1736–1751, 2020.

[101] S. Vahidian, M. Morafah, C. Chen, M. Shah, and B. Lin, "Rethinking data heterogeneity in federated learning: Introducing a new notion and standard benchmarks," *IEEE Trans. Artif. Intell.*, vol. 5, no. 3, pp. 1386–1397, Mar. 2024.

[102] Y. Dai, Z. Chen, J. Li, S. Heinecke, L. Sun, and R. Xu, "Tackling data heterogeneity in federated learning with class prototypes," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 7314–7322.

[103] D. Liao, X. Gao, Y. Zhao, and C.-Z. Xu, "Adaptive channel sparsity for federated learning under system heterogeneity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 20432–20441.

[104] K. Pfeiffer, M. Rapp, R. Khalili, and J. Henkel, "Federated learning for computationally constrained heterogeneous devices: A survey," *ACM Comput. Surv.*, vol. 55, no. 14s, pp. 1–27, 2023.

[105] J. Zhang et al., "FedALA: Adaptive local aggregation for personalized federated learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 11237–11244.

[106] D. Chen, L. Yao, D. Gao, B. Ding, and Y. Li, "Efficient personalized federated learning via sparse model-adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 5234–5256.

[107] Z. Qin, L. Yang, Q. Wang, Y. Han, and Q. Hu, "Reliable and interpretable personalized federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 20422–20431.

[108] A. Vettoruzzo, M.-R. Bouguelia, J. Vanschoren, T. Rognvaldsson, and K. Santosh, "Advances and challenges in meta-learning: A technical review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 7, pp. 4763–4779, Jul. 2024.

[109] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Trans. Knowl. data Eng.*, vol. 34, no. 12, pp. 5586–5609, Dec. 2022.

[110] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15262–15271.

[111] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 819–828.

[112] C. Li et al., "An embarrassingly simple backdoor attack on self-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4367–4378.

[113] J. Xue et al., "TrojLLM: A black-box trojan prompt attack on large language models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–3.

[114] Z. Tian, L. Cui, J. Liang, and S. Yu, "A comprehensive survey on poisoning attacks and countermeasures in machine learning," *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–35, 2022.

[115] Z. Yang et al., "Data poisoning attacks against multimodal encoders," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 39299–39313.

[116] J. Zhang, S. Peng, Y. Gao, Z. Zhang, and Q. Hong, "APMSA: Adversarial perturbation against model stealing attacks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1667–1679, 2023.

[117] S. Sanyal, S. Addepalli, and R. V. Babu, "Towards data-free model stealing in a hard label setting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15284–15293.

[118] K. Yoo and N. Kwak, "Backdoor attacks in federated learning by rare embeddings and gradient ensembling," 2022, *arXiv:2204.14017*.

[119] Y. Chen, W. Lu, X. Qin, J. Wang, and X. Xie, "MetaFed: Federated learning among federations with cyclic knowledge distillation for personalized healthcare," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 16671–16682, Nov. 2024.

[120] J. Zhu, J. Cao, D. Saxena, S. Jiang, and H. Ferradi, "Blockchain-empowered federated learning: Challenges, solutions, and future directions," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–31, 2023.

[121] J. Wang et al., "Towards personalized federated learning via heterogeneous model reassembly," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 29515–29531.

[122] H. Li et al., "FedTP: Federated learning by transformer personalization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 13426–13440, Oct. 2024.

[123] A. Hard et al., "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[124] T. Yang et al., "Applied federated learning: Improving Google keyboard query suggestions," 2018, *arXiv:1812.02903*.

[125] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.

[126] D. Zeng, S. Liang, X. Hu, H. Wang, and Z. Xu, "FedLab: A flexible federated learning framework," *J. Mach. Learn. Res.*, vol. 24, no. 100, pp. 1–7, 2023.

[127] Q. Zhang, T. Wu, P. Zhou, S. Zhou, Y. Yang, and X. Jin, "Felicitas: Federated learning in distributed cross device collaborative frameworks," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2022, pp. 4502–4509.

[128] H. Ludwig et al., "IBM federated learning: An enterprise framework white paper v0. 1," 2020, *arXiv:2007.10987*.

[129] Y. Ma, D. Yu, T. Wu, and H. Wang, "PaddlePaddle: An open-source deep learning platform from industrial practice," *Front. Data Domput.*, vol. 1, no. 1, pp. 105–115, 2019.

[130] C. He et al., "FedML: A research library and benchmark for federated machine learning," 2020, *arXiv:2007.13518*.

[131] B. Y. Lin et al., "FedNLP: Benchmarking federated learning methods for natural language processing tasks," 2021, *arXiv:2104.08815*.

[132] T. Fan et al., "FATE-LLM: A industrial grade federated learning framework for large language models," 2023, *arXiv:2310.10049*.

[133] W. Kuang et al., "Federatedscope-LLM: A comprehensive package for fine-tuning large language models in federated learning," 2023, *arXiv:2309.00363*.

[134] R. Ye et al., "OpenFedLLM: Training large language models on Decentralized private data via federated learning," 2024, *arXiv:2402.06954*.

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

[136] J. Zhao, "Privacy-preserving fine-tuning of artificial intelligence (AI) foundation models with federated learning, differential privacy, Offsite tuning, and parameter-efficient fine-tuning (PEFT)," TechRxiv, Preprints, 2023.

[137] D. J. Beutel et al., "Flower: A friendly federated learning research framework," 2020, *arXiv:2007.14390*.

[138] W. Lu, X. Hu, J. Wang, and X. Xie, "FedCLIP: Fast generalization and personalization for CLIP in federated learning," 2023, *arXiv:2302.13485*.

[139] Y. Gao, J. Fernandez-Marques, T. Parcollet, A. Mehrotra, and N. D. Lane, "Federated self-supervised speech representations: Are we there yet?" 2022, *arXiv:2204.02804*.

[140] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12449–12460.

[141] S. Moon, T. S. Kim, J. Ryu, and W. H. Lee, "Federated learning for sleep stage classification on edge devices via a model-agnostic Meta-learning-based pre-trained model," in *Proc. IEEE 13th Int. Conf. Consum. Electron.*, 2023, pp. 188–192.

[142] Y. Tan, G. Long, J. Ma, L. Liu, T. Zhou, and J. Jiang, "Federated learning from pre-trained models: A contrastive learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 19332–19344.

[143] J. Nguyen, J. Wang, K. Malik, M. Sanjabi, and M. Rabbat, "Where to begin? On the impact of pre-training and initialization in federated learning," 2022, *arXiv:2210.08090*.

[144] H.-Y. Chen, C.-H. Tu, Z. Li, H. W. Shen, and W.-L. Chao, "On the importance and applicability of pre-training for federated learning," in *Proc. 11th Int. Conf. Learn. Represent.*, 2022, pp. 1–26.

[145] Z. Charles, N. Mitchell, K. Pillutla, M. Reneer, and Z. Garrett, "Towards federated foundation models: Scalable dataset pipelines for group-structured learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–29.

[146] T. Zhang, T. Feng, S. Alam, M. Zhang, S. S. Narayanan, and S. Avestimehr, "GPT-FL: Generative pre-trained model-assisted federated learning," 2023, *arXiv:2306.02210*.

[147] B. Wang et al., "Can public large language models help private cross-device federated learning?" 2023, *arXiv:2305.12132*.

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

[149] S. Malaviya, M. Shukla, and S. Lodha, "Reducing communication overhead in federated learning for pre-trained language models using parameter-efficient finetuning," in *Proc. Conf. Lifelong Learn. Agents*, 2023, pp. 456–469.

[150] J. Chen, W. Xu, S. Guo, J. Wang, J. Zhang, and H. Wang, "FedTune: A deep dive into efficient federated fine-tuning with pre-trained transformers," 2022, *arXiv:2211.08025*.

[151] Z. Zhang et al., "FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models," in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, 2023, pp. 9963–9977.

[152] H. Zhao, W. Du, F. Li, P. Li, and G. Liu, "FedPrompt: Communication-efficient and privacy-preserving prompt tuning in federated learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.

[153] T. Guo, S. Guo, J. Wang, X. Tang, and W. Xu, "PromptFL: Let federated participants cooperatively learn prompts instead of models-federated learning in age of foundation model," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5179–5194, May 2024.

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

[155] P. Riedel, M. Reichert, R. Von Schwerin, A. Hafner, D. Schaudt, and G. Singh, "Performance analysis of federated learning algorithms for multilingual protest news detection using pre-trained DistilBERT and BERT," *IEEE Access*, vol. 11, pp. 134009–134022, 2023.

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

[157] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, and J. Lin, "Distilling task-specific knowledge from BERT into simple neural networks," 2019, *arXiv:1903.12136*.

[158] J. Shin et al., "FedTherapist: Mental health monitoring with user-generated linguistic expressions on smartphones via federated learning," 2023, *arXiv:2310.16538*.

[159] L. Yue, Q. Liu, Y. Du, W. Gao, Y. Liu, and F. Yao, "FedJudge: Federated legal large language model," 2023, *arXiv:2309.08173*.

[160] D. Sui, Y. Chen, J. Zhao, Y. Jia, Y. Xie, and W. Sun, "FedED: Federated learning via ensemble distillation for medical relation extraction," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2020, pp. 2118–2128.

[161] U. Ahmed, J. C.-W. Lin, and G. Srivastava, "Semisupervised federated learning for temporal news hyperpatism detection," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 4, pp. 1758–1769, Aug. 2023.

[162] X. Guo et al., "Federated learning for personalized humor recognition," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–18, 2022.

[163] J. Yi, F. Wu, C. Wu, R. Liu, G. Sun, and X. Xie, "Efficient-FedRec: Efficient federated learning framework for privacy-preserving news recommendation," 2021, *arXiv:2109.05446*.

[164] W. Dong, X. Wu, J. Li, S. Wu, C. Bian, and D. Xiong, "FewFedWeight: Few-shot federated learning framework across multiple NLP tasks," 2022, *arXiv:2212.08354*.

[165] D. Cai, Y. Wu, S. Wang, F. X. Lin, and M. Xu, "Efficient federated learning for modern NLP," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, 2023, pp. 1–16.

[166] Y. Jiang et al., "Model pruning enables efficient federated learning on edge devices," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10374–10386, Dec. 2023.

[167] T. Che et al., "Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization," 2023, *arXiv:2310.15080*.

[168] L. Collins, S. Wu, S. Oh, and K. C. Sim, "Profit: Benchmarking personalization and robustness trade-off in federated prompt tuning," 2023, *arXiv:2310.04627*.

[169] H. Chen, Y. Zhang, D. Krompass, J. Gu, and V. Tresp, "FedDAT: An approach for foundation model Finetuning in multi-modal heterogeneous federated learning," 2023, *arXiv:2308.12305*.

[170] S. Su, B. Li, and X. Xue, "FedRA: A random allocation strategy for federated tuning to unleash the power of heterogeneous clients," 2023, *arXiv:2311.11227*.

[171] L. Yi, H. Yu, G. Wang, and X. Liu, "FedLoRA: Model-heterogeneous personalized federated learning with LoRa tuning," 2023, *arXiv:2310.13283*.

[172] F.-E. Yang, C.-Y. Wang, and Y.-C. F. Wang, "Efficient model personalization in federated learning via client-specific prompt generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 19159–19168.

[173] Y. J. Cho, L. Liu, Z. Xu, A. Fahrezi, M. Barnes, and G. Joshi, "Heterogeneous LoRA for federated fine-tuning of on-device foundation models," in *Proc. Int. Workshop Federated Learn. Age Found. Models Conjunct. NeurIPS*, 2023, pp. 1–8.

[174] C. Xie et al., "PerAda: Parameter-efficient and Generalizable federated learning personalization with guarantees," 2023, *arXiv:2302.06637*.

[175] Y. Tian, Y. Wan, L. Lyu, D. Yao, H. Jin, and L. Sun, "FedBERT: When federated learning meets pre-training," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–26, 2022.

[176] H. L. Xin'ao Wang, K. Chen, and L. Shou, "FedBFPT: An efficient federated learning framework for BERT further pre-training," in *Proc. 32nd Int. Joint Conf. Artif. Intell.*, 2023, pp. 4344–4352.

[177] Y. Chen, Z. Chen, P. Wu, and H. Yu, "FedOBD: Opportunistic block dropout for efficiently training large-scale neural networks through federated learning," 2022, *arXiv:2208.05174*.

[178] G. Sun, M. Mendieta, J. Luo, S. Wu, and C. Chen, "Fedperfix: Towards partial model personalization of vision transformers in federated learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4988–4998.

[179] C. Xu and J. McAuley, "A survey on model compression and acceleration for pretrained language models," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 10566–10575.

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

[181] L. Li, Y. Zhang, and L. Chen, "Prompt distillation for efficient LLM-based recommendation," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, 2023, pp. 1348–1357.

[182] M. Kang, S. Lee, J. Baek, K. Kawaguchi, and S. J. Hwang, "Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks," in *Proc. 37th Int. Conf. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 48573–48602.

[183] S. Yu, J. P. Muñoz, and A. Jannesari, "Bridging the gap between foundation models and heterogeneous federated learning," 2023, *arXiv:2310.00247*.
[184] Y. J. Cho, A. Manoel, G. Joshi, R. Sim, and D. Dimitriadis, "Heterogeneous ensemble knowledge transfer for training large models in federated learning," 2022, *arXiv:2204.12703*.
[185] Y. Deng, J. Ren, C. Tang, F. Lyu, Y. Liu, and Y. Zhang, "A hierarchical knowledge transfer framework for heterogeneous federated learning," in *Proc. INFOCOM IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.
[186] D. Yao et al., "FedGKD: Towards heterogeneous federated learning via global knowledge distillation," *IEEE Trans. Comput.*, vol. 73, no. 1, pp. 3–17, Jan. 2024.
[187] H. Mei, D. Cai, Y. Wu, S. Wang, and M. Xu, "A survey of backpropagation-free training for LLMS," TechRxiv, Preprint, 2024. [Online]. Available: http://dx.doi.org/10.36227/techrxiv.171172909.97532161/v1
[188] D. P. Pau and F. M. Aymone, "Suitability of forward-forward and PEPITA learning to MLCommons-tiny benchmarks," in *Proc. IEEE Int. Conf. Omni-Layer Intell. Syst. (COINS)*, 2023, pp. 1–6.
[189] S. Malladi et al., "Fine-tuning language models with just forward passes," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 53038–53075.
[190] M. Xu, Y. Wu, D. Cai, X. Li, and S. Wang, "Federated fine-tuning of billion-sized language models across mobile devices," 2023, *arXiv:2308.13894*.
[191] W. Lu et al., "ZooPFL: Exploring black-box foundation models for personalized federated learning," 2023, *arXiv:2310.05143*.
[192] J. Sun et al., "FedBPT: Efficient federated black-box prompt tuning for large language models," 2023, *arXiv:2310.01467*.
[193] Z. Qin, D. Chen, B. Qian, B. Ding, Y. Li, and S. Deng, "Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes," 2023, *arXiv:2312.06353*.
[194] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 691–706.
[195] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 14774–14784.
[196] H. Yang, M. Ge, D. Xue, K. Xiang, H. Li, and R. Lu, "Gradient leakage attacks in federated learning: Research frontiers, taxonomy, and future directions," *IEEE Netw.*, vol. 38, no. 2, pp. 247–254, Mar. 2024.
[197] W. Zhang, S. Tople, and O. Ohrimenko, "Leakage of dataset properties in multi-party machine learning," in *Proc. 30th USENIX Secur. Symp. (USENIX Secur.)*, 2021, pp. 2687–2704.
[198] J. Zhao, H. Zhu, F. Wang, R. Lu, and H. Li, "Efficient and privacy-preserving tree-based inference via additive homomorphic encryption," *Inf. Sci.*, vol. 650, Dec. 2023, Art. no. 119480.
[199] H. Liu, B. Li, P. Xie, and C. Zhao, "Privacy-encoded federated learning against gradient-based data reconstruction attacks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 5860–5875, 2023.
[200] X. Pan, M. Zhang, S. Ji, and M. Yang, "Privacy risks of general-purpose language models," in *Proc. IEEE Symp. Security Privacy (SP)*, 2020, pp. 1314–1331.
[201] J. Jeon, K. Lee, K. Lee, S. Oh, and J. Ok, "Gradient inversion with generative image prior," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 29898–29908.
[202] W. Wei, L. Liu, Y. Wu, G. Su, and A. Iyengar, "Gradient-leakage resilient federated learning," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2021, pp. 797–807.
[203] K. Yue, R. Jin, C.-W. Wong, D. Baron, and H. Dai, "Gradient obfuscation gives a false sense of security in federated learning," in *Proc. 32nd USENIX Secur. Symp. (USENIX Secur.)*, 2023, pp. 6381–6398.
[204] B. C. Das, M. H. Amini, and Y. Wu, "Privacy risks analysis and mitigation in federated learning for medical images," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, 2023, pp. 1870–1873.
[205] J. Deng et al., "Tag: Gradient attack on transformer-based language models," 2021, *arXiv:2103.06819*.
[206] M. Balunovic, D. Dimitrov, N. Jovanović, and M. Vechev, "LAMP: Extracting text from gradients with language model priors," in *Proc. 36th Conf. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 7641–7654.
[207] S. Gupta, Y. Huang, Z. Zhong, T. Gao, K. Li, and D. Chen, "Recovering private text in federated learning of language models," in *Proc. 36th Conf. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 8130–8143.
[208] L. Fowl et al., "Decepticons: Corrupted transformers breach privacy in federated learning for language models," 2022, *arXiv:2201.12675*.
[209] Q. Xu, J. Wang, O. Ohrimenko, and T. Cohn, "FLAT-ChaT: A word recovery attack on federated language model training," 2023, submitted for publication.
[210] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.
[211] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 739–753.
[212] V. Shejwalkar, H. A. Inan, A. Houmansadr, and R. Sim, "Membership inference attacks against NLP classification models," in *Proc. 35th Conf. Workshop Privacy Mach. Learn. NeurIPS*, 2021, pp. 1–13.
[213] Y. Wang et al., "Analyzing and defending against membership inference attacks in natural language processing classification," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2022, pp. 5823–5832.
[214] J. Hauser, Z. Meng, D. Pascual, and R. Wattenhofer, "Bert is robust! A case against word substitution-based adversarial attacks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.
[215] M. Bertran, S. Tang, A. Roth, M. Kearns, J. H. Morgenstern, and S. Z. Wu, "Scalable membership inference attacks via quantile regression," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–17.
[216] M. Ko, M. Jin, C. Wang, and R. Jia, "Practical membership inference attacks against large-scale multi-modal models: A pilot study," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4871–4881.
[217] C. Song and A. Raghunathan, "Information leakage in embedding models," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 377–390.
[218] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[219] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 2016, *arXiv:1607.01759*.
[220] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empir. Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[221] M. Mahoney. "Large text compression benchmark." 2009. [Online]. Available: https://cs.fit.edu/ mmahoney/compression/text.html
[222] D. Cer et al., "Universal sentence encoder for English," in *Proc. Conf. Empir. Methods Natural Lang. Process., Syst. Demonstrat.*, 2018, pp. 169–174.
[223] F. Mireshghallah, K. Goyal, A. Uniyal, T. Berg-Kirkpatrick, and R. Shokri, "Quantifying privacy risks of masked language models using membership inference attacks," 2022, *arXiv:2203.03929*.
[224] T. Vakili and H. Dalianis, "Using membership inference attacks to evaluate privacy-preserving language modeling fails for pseudonymizing data," in *Proc. 24th Nordic Conf. Comput. Linguist. (NoDaLiDa)*, 2023, pp. 318–323.
[225] A. Jagannatha, B. P. S. Rawat, and H. Yu, "Membership inference attack susceptibility of clinical language models," 2021, *arXiv:2104.08305*.
[226] C. Song and V. Shmatikov, "Auditing data provenance in text-generation models," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, 2019, pp. 196–206.
[227] M. Meeus, S. Jain, M. Rei, and Y.-A. de Montjoye, "Did the neurons read your book? Document-level membership inference for large language models," 2023, *arXiv:2310.15007*.
[228] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *Proc. IEEE Symp. Security Privacy (SP)*, 2022, pp. 1897–1914.
[229] J. Abascal, S. Wu, A. Oprea, and J. Ullman, "TMI! Finetuned models leak private information from their pretraining data," 2023, *arXiv:2306.01181*.
[230] R. Staab, M. Vero, M. Balunović, and M. Vechev, "Beyond memorization: Violating privacy via inference with large language models," 2023, *arXiv:2310.07298*.
[231] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. 3rd Theory Cryptogr. Conf., (TCC)*, New York, NY, USA, 2006, pp. 265–284.
[232] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.
[233] A. Dubey and A. Pentland, "Differentially-private federated linear bandits," in *Proc. 34th. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6003–6014.
[234] M. Letafati and S. Otoum, "Global differential privacy for distributed metaverse healthcare systems," in *Proc. Int. Conf. Intell. Metaverse Technol. Appl. (iMETA)*, 2023, pp. 1–8.

[235] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6532–6542, Oct. 2020.

[236] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.

[237] H. Cao, S. Liu, R. Zhao, and X. Xiong, "IFed: A novel federated learning framework for local differential privacy in power Internet of Things," *Int. J. Distrib. Sensor Netw.*, vol. 16, no. 5, 2020, Art. no. 1550147720919698.

[238] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 2351–2363.

[239] M. Yang, T. Guo, T. Zhu, I. Tjuawinata, J. Zhao, and K.-Y. Lam, "Local differential privacy and its applications: A comprehensive survey," *Comput. Stand. Interfaces*, vol. 89, Apr. 2023, Art. no. 103827.

[240] Z. Xu et al., "Learning to generate image embeddings with user-level differential privacy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7969–7980.

[241] V. A. Farias, F. T. Brito, C. Flynn, J. C. Machado, S. Majumdar, and D. Srivastava, "Local dampening: Differential privacy for non-numeric queries via local sensitivity," *VLDB J.*, vol. 32, pp. 1191–1214, Nov. 2023.

[242] L. Hu et al., "Defenses to membership inference attacks: A survey," *ACM Comput. Surveys*, vol. 56, no. 4, pp. 1–34, 2023.

[243] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends® Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[244] P. Basu, T. S. Roy, R. Naidu, Z. Muftuoglu, S. Singh, and F. Mireshghallah, "Benchmarking differential privacy and federated learning for bert models," 2021, *arXiv:2106.13973*.

[245] P. Basu, T. S. Roy, R. Naidu, and Z. Muftuoglu, "Privacy enabled financial text classification using differential privacy and federated learning," 2021, *arXiv:2110.01643*.

[246] X.-Y. Liu et al., "Differentially private low-rank adaptation of large language model using federated learning," 2023, *arXiv:2312.17493*.

[247] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*, 1999, pp. 223–238.

[248] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.

[249] A. C. Yao, "Protocols for secure computations," in *Proc. 23rd Annu. Symp. Found. Comput. Sci. (SFCS)*, 1982, pp. 160–164.

[250] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. 27th Annu. Symp. Found. Comput. Sci. (SFCS)*, 1986, pp. 162–167.

[251] M. O. Rabin, "How to exchange secrets with oblivious transfer," Cryptol. ePrint Arch., IACR, Bellevue, WA, USA, Rep. 2005/187, 2005.

[252] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[253] S. Goryczka and L. Xiong, "A comprehensive comparison of multiparty secure additions with differential privacy," *IEEE Trans. Depend. Secure Comput.*, vol. 14, no. 5, pp. 463–477, Sep./Oct. 2015.

[254] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2010, pp. 735–746.

[255] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3733–3744, Aug. 2018.

[256] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. 29th USENIX Secur. Symp. (USENIX Secur.)*, 2020, pp. 1605–1622.

[257] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2018, pp. 19–35.

[258] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation," in *Proc. 35th Uncertainty Artif. Intell.*, 2020, pp. 261–270.

[259] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2938–2948.

[260] Z. Cheng, B. Wu, Z. Zhang, and J. Zhao, "TAT: Targeted backdoor attacks against visual object tracking," *Pattern Recognit.*, vol. 142, Oct. 2023, Art. no. 109629.

[261] Y. Gao, Y. Li, L. Zhu, D. Wu, Y. Jiang, and S.-T. Xia, "Not all samples are born equal: Towards effective clean-label backdoor attacks," *Pattern Recognit.*, vol. 139, Jul. 2023, Art. no. 109512.

[262] W. Feng, N. Xu, T. Zhang, and Y. Zhang, "Dynamic generative targeted attacks with pattern injection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 16404–16414.

[263] W. Wan et al., "A four-pronged defense against Byzantine attacks in federated learning," in *Proc. 31st ACM Int. Conf. Multimedia*, 2023, pp. 7394–7402.

[264] J. Shi, W. Wan, S. Hu, J. Lu, and L. Y. Zhang, "Challenges and approaches for mitigating Byzantine attacks in federated learning," in *Proc. IEEE Int. Conf. Trust, Security Privacy Comput. Commun. (TrustCom)*, 2022, pp. 139–146.

[265] H. Wei, H. Zhang, A.-H. Kamal, and Y. Shi, "Ensuring secure platooning of constrained intelligent and connected vehicles against Byzantine attacks: A distributed MPC framework," *Engineering*, vol. 33. pp. 35–46, Feb. 2023.

[266] Z. Zhang and R. Hu, "Byzantine-robust federated learning with variance reduction and differential privacy," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2023, pp. 1–9.

[267] Z. Wu, Q. Ling, T. Chen, and G. B. Giannakis, "Federated variance-reduced stochastic gradient descent with robustness to Byzantine attacks," *IEEE Trans. Signal Process.*, vol. 68, pp. 4583–4596, 2020.

[268] S. Li, E. C.-H. Ngai, and T. Voigt, "An experimental study of Byzantine-robust aggregation schemes in federated learning," *IEEE Trans. Big Data*, vol. 10, no. 6, pp. 975–988, Dec. 2024.

[269] C. Xie, S. Koyejo, and I. Gupta, "Zeno++: Robust fully asynchronous SGD," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10495–10503.

[270] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[271] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 3521–3530.

[272] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Min.*, 2022, pp. 2545–2555.

[273] Z. Gu and Y. Yang, "Detecting malicious model updates from federated learning on conditional variational autoencoder," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, 2021, pp. 671–680.

[274] X. Zhang et al., "Secure collaborative learning in mining pool via robust and efficient verification," in *Proc. IEEE 43rd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2023, pp. 794–805.

[275] B. Nelson et al., "Exploiting machine learning to subvert your spam filter," in *Proc. LEET*, 2008, pp. 16–17.

[276] J. Newsome, B. Karp, and D. Song, "Paragraph: Thwarting signature learning by training maliciously," in *Proc. 9th Int. Symp. Recent Adv. Intrusion Detect. (RAID)*, Hamburg, Germany, 2006, pp. 81–105.

[277] B. I. Rubinstein et al., "Antidote: Understanding and defending against poisoning of anomaly detectors," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas.*, 2009, pp. 1–14.

[278] C. Schuhmann et al., "LAION-5B: An open large-scale dataset for training next generation image-text models," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 25278–25294.

[279] M. Byeon, B. Park, H. Kim, S. Lee, W. Baek, and S. Kim. "COYO-700M: Image-text pair dataset." 2022. [Online]. Available: https://github.com/kakaobrain/coyo-dataset

[280] N. Carlini et al., "Poisoning Web-scale training datasets is practical," 2023, *arXiv:2302.10149*.

[281] N. Carlini, "Poisoning the unlabeled dataset of semi-supervised learning," in *Proc. 30th USENIX Secur. Symp. (USENIX Secur.)*, 2021, pp. 1577–1592.

[282] Z. Yang et al., "Stealthy backdoor attack for code models," *IEEE Trans. Softw. Eng.*, vol. 50, no. 4, pp. 721–741, Feb. 2024.

[283] N. Kandpal, M. Jagielski, F. Tramèr, and N. Carlini, "Backdoor attacks for in-context learning with language models," 2023, *arXiv:2307.14692*.

[284] Y. Yuan, R. Kong, S. Xie, Y. Li, and Y. Liu, "Patchbackdoor: Backdoor attack against deep neural networks without model modification," in *Proc. 31st ACM Int. Conf. Multimedia*, 2023, pp. 9134–9142.

[285] S. Han et al., "FedMLSecurity: A benchmark for attacks and defenses in federated learning and LLMs," 2023, *arXiv:2306.04959*.

[286] X. Li, C. Wu, and J. Wang, "Unveiling backdoor risks brought by foundation models in heterogeneous federated learning," 2023, *arXiv:2311.18350*.
[287] X. Li, S. Wang, C. Wu, H. Zhou, and J. Wang, "Backdoor threats from compromised foundation models to federated learning," in *Proc. 37th Int. Workshop Federated Learn. Age Found. Models Conjunct. NeurIPS*, 2023, pp. 1–12.
[288] C. Wu, X. Li, and J. Wang, "Vulnerabilities of foundation model integrated federated learning under adversarial threats," 2024, *arXiv:2401.10375*.
[289] T. Zhou, H. Yan, B. Han, L. Liu, and J. Zhang, "Learning a robust foundation model against clean-label data poisoning attacks at downstream tasks," *Neural Netw.*, vol. 169, pp. 756–763, Jan. 2024.
[290] W. Huang, Y. Wang, A. Cheng, A. Zhou, C. Yu, and L. Wang, "A fast, Performant, secure distributed training framework for LLM," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2024, pp. 4800–4804.
[291] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in *Proc. 16th ACM Workshop Artif. Intell. Secur.*, 2023, pp. 79–90.
[292] P. Rai, S. Sood, V. K. Madisetti, and A. Bahga, "GUARDIAN: A multi-tiered defense architecture for thwarting prompt injection attacks on LLMs," *J. Softw. Eng. Appl.s*, vol. 17, no. 1, pp. 43–68, 2024.
[293] J. Yan et al., "Backdooring instruction-tuned large language models with virtual prompt injection," in *Proc. NeurIPS Workshop Backdoors Deep, Learn.-Good, Bad, Ugly*, 2023, pp. 1–19.
[294] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does LLM safety training fail?" in *Proc. 37th Conf. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–32.
[295] J. Wang et al., "On the robustness of ChatGPT: An adversarial and out-of-distribution perspective," 2023, *arXiv:2302.12095*.
[296] N. Carlini et al., "Extracting training data from large language models," in *Proc. 30th USENIX Secur. Symp. (USENIX Secur.)*, 2021, pp. 2633–2650.
[297] Z. Tan and M. Jiang, "User modeling in the era of large language models: Current research and future directions," 2023, *arXiv:2312.11518*.
[298] R. Koike, M. Kaneko, and N. Okazaki, "Outfox: LLM-generated essay detection through in-context learning with adversarially generated examples," 2023, *arXiv:2307.11729*.
[299] Y. Liu, Y. Jia, R. Geng, J. Jia, and N. Z. Gong, "Prompt injection attacks and defenses in LLM-integrated applications," 2023, *arXiv:2310.12815*.
[300] F. Wang, E. Hugh, and B. Li, "More than enough is too much: Adaptive defenses against gradient leakage in production federated learning," *IEEE/ACM Trans. Netw.*, vol. 32, no. 4, pp. 3061–3075, Aug. 2024.
[301] R. Dwivedi et al., "Explainable AI (XAI): Core ideas, techniques, and solutions," *ACM Comput. Surveys*, vol. 55, no. 9, pp. 1–33, 2023.
[302] V. Chamola, V. Hassija, A. R. Sulthana, D. Ghosh, D. Dhingra, and B. Sikdar, "A review of trustworthy and Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 11, pp. 78994–79015, 2023.
[303] A. Roy, M. Saffar, A. Vaswani, and D. Grangier, "Efficient content-based sparse attention with routing transformers," *Trans. Assoc. Comput. Linguist.*, vol. 9, pp. 53–68, Feb. 2021.
[304] S. Dai, H. Genc, R. Venkatesan, and B. Khailany, "Efficient transformer inference with statically structured sparse attention," in *Proc. 60th ACM/IEEE Design Autom. Conf. (DAC)*, 2023, pp. 1–6.
[305] B. Li et al., "DyStyle: Dynamic neural network for multi-attribute-conditioned style Editings," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 189–197.
[306] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7436–7456, Nov. 2022.
[307] W. Cui et al., "Optimizing dynamic neural networks with brainstorm," in *Proc. 17th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2023, pp. 797–815.
[308] E. Frantar and D. Alistarh, "SparseGPT: Massive language models can be accurately pruned in one-shot," in *Proc. 40th Int. Conf. Mach. Learn.*, 2023, pp. 1–15.
[309] X. Ma, G. Fang, and X. Wang, "LLM-Pruner: On the structural pruning of large language models," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 21702–21720.
[310] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLORA: Efficient finetuning of quantized LLMs," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–28.
[311] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "Smoothquant: Accurate and efficient post-training quantization for large language models," in *Proc. 40th Int. Conf. Mach. Learn.*, 2023, pp. 38087–38099.
[312] Z. Zheng, X. Ren, F. Xue, Y. Luo, X. Jiang, and Y. You, "Response length perception and sequence scheduling: An LLM-empowered LLM inference pipeline," in *Proc. 37th Int. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 65517–65530.
[313] Z. Liu et al., "Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–23.
[314] T. Zheng, A. Li, Z. Chen, H. Wang, and J. Luo, "Autofed: Heterogeneity-aware federated multimodal learning for robust autonomous driving," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, 2023, pp. 1–15.
[315] R. Islam and O. M. Moushi, "GPT-4o: The cutting-edge advancement in multimodal LLM," TechRxiv, Preprint, 2024. [Online]. Available: http://dx.doi.org/10.36227/techrxiv.171986596.65533294/v1
[316] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[317] R. Girdhar et al., "Imagebind: One embedding space to bind them all," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 15180–15190.
[318] Y. Li et al., "Personal LLM agents: Insights and survey about the capability, efficiency and security," 2024, *arXiv:2401.05459*.
[319] D. Shenaj, M. Toldo, A. Rigon, and P. Zanuttigh, "Asynchronous federated continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 5054–5062.
[320] P. Fernandez, A. Chaffin, K. Tit, V. Chappelier, and T. Furon, "Three bricks to consolidate watermarks for large language models," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, 2023, pp. 1–6.
[321] Q. Pang, S. Hu, W. Zheng, and V. Smith, "Attacking LLM watermarks by exploiting their strengths," 2024, *arXiv:2402.16187*.
[322] Y. Qu, M. Ding, N. Sun, K. Thilakarathna, T. Zhu, and D. Niyato, "The frontier of data erasure: Machine unlearning for large language models," 2024, *arXiv:2403.15779*.
[323] Z. Hu, Y. Zhang, M. Xiao, W. Wang, F. Feng, and X. He, "Exact and efficient unlearning for large language model-based recommendation," 2024, *arXiv:2404.10327*.
[324] V. Patil, P. Hase, and M. Bansal, "Can sensitive information be deleted from LLMS? Objectives for defending against extraction attacks," 2023, *arXiv:2309.17410*.
[325] X. Qiu, W. F. Shen, Y. Chen, N. Cancedda, P. Stenetorp, and N. D. Lane, "PISTOL: Dataset compilation pipeline for structural unlearning of LLMs," 2024, *arXiv:2406.16810*.

Yujun Cheng received the B.S. and Ph.D. degrees from the Department of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China, in 2014 and 2019, respectively. He is currently a Postdoctoral Researcher with the Department of Electronic Engineering, Tsinghua University, Beijing. He has published research papers in top-tier communication and computer science conferences and journals, including IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and IEEE INTERNET OF THINGS JOURNAL. His research interests include Internet of Things, edge computing, federated learning, and machine learning.

Weiting Zhang (Member, IEEE) received the Ph.D. degree in communication and information systems from the Beijing Jiaotong University, Beijing, China, in 2021. From November 2019 to November 2020, he was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Since December 2021, he has been working as an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include industrial Internet of Things, deterministic networks, edge intelligence, and machine learning for network optimization.

Zhewei Zhang received the B.S. and Ph.D. degrees from Beijing Jiaotong University, Beijing, China, in 2014 and 2019, respectively. He is currently a Postdoctoral Fellow with the Department of Electronics, Tsinghua University, Beijing. He has published research papers in top computer science conferences and journals, such as IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. His main research interests include artificial intelligence, computer vision, machine learning algorithms, pattern recognition, adversarial generative networks, and diffusion models. He was awarded the Google Scholarship in 2018.

Chuan Zhang (Member, IEEE) received the Ph.D. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2021. From September 2019 to September 2020, he worked as a visiting Ph.D. student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently an Assistant Professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include secure data services in cloud computing, applied cryptography, machine learning, and blockchain.

Shengjin Wang (Senior Member, IEEE) received the B.E. degree from Tsinghua University, China, in 1985, and the Ph.D. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 1997. From May 1997 to August 2003, he was a member of the Research Staff with the Internet System Research Laboratories, NEC Corporation, Japan. Since September 2003, he has been a Professor with the Department of Electronic Engineering, Tsinghua University, where he is currently also the Director of the Research Center for Media Big-Data Cognitive Computing. He has published more than 100 articles and possessed more than 20 patents. His current research interests include image processing, computer vision, multimodal cooperative robot, and person re-identification.

Shiwen Mao (Fellow, IEEE) is a Professor and Earle C. Williams Eminent Scholar, and the Director of the Wireless Engineering Research and Education Center, Auburn University. His research interest includes wireless networks, multimedia communications, and smart grid. He received the IEEE ComSoc MMTC Outstanding Researcher Award in 2023, the Southeastern Conference 2023 Faculty Achievement Award for Auburn, the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award in 2019, the Auburn University Creative Research and Scholarship Award in 2018, and the NSF CAREER Award in 2010, and several service awards from IEEE ComSoc. He is a co-recipient of the 2022 Best Journal Paper Award of IEEE ComSoc eHealth Technical Committee, the 2021 Best Paper Award of Elsevier/KeAi Digital Communications and Networks Journal, the 2021 IEEE INTERNET OF THINGS JOURNAL Best Paper Award, the 2021 IEEE Communications Society Outstanding Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the 2018 Best Journal Paper Award and the 2017 Best Conference Paper Award from IEEE ComSoc MMTC, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is a co-recipient of the Best Paper Awards from GLOBECOM 2023 (two), 2019, 2016, and 2015, IEEE ICC 2022 and 2013, and IEEE WCNC 2015, and the Best Demo Awards from IEEE INFOCOM 2022 and IEEE SECON 2017. He is the Editor-in-Chief of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is a Distinguished Lecturer of IEEE Communications Society from 2021 to 2025, the IEEE Council of RFID from 2021 to 2023, and the IEEE Vehicular Technology Society (VTS) from 2014 to 2018, and a Distinguished Speaker of IEEE VTS from 2018 to 2021. He was the General Chair of IEEE INFOCOM 2022, the TPC Chair of IEEE INFOCOM 2018, and the TPC Vice-Chair of IEEE GLOBECOM 2022.