Federated Large Language Models: Feasibility, Robustness, Security and Future Directions

WENHAO JIANG, National University of Defense Technology, Changsha, China
YUCHUAN LUO, National University of Defense Technology, Changsha, China
GUILIN DENG, National University of Defense Technology, Changsha, China
SILONG CHEN, National University of Defense Technology, Changsha, China
XU YANG, National University of Defense Technology, Changsha, China
SHIHONG WU, National University of Defense Technology, Changsha, China
XINWEN GAO, National University of Defense Technology, Changsha, China
LIN LIU, National University of Defense Technology, Changsha, China
SHAOJING FU, National University of Defense Technology, Changsha, China

The integration of Large Language Models (LLMs) and Federated Learning (FL) presents a promising solution for joint training on distributed data while preserving privacy and addressing data silo issues. However, this emerging field, known as Federated Large Language Models (FLLM), faces significant challenges, including communication and computation overheads, heterogeneity, privacy and security concerns. Current research has primarily focused on the feasibility of FLLM, but future trends are expected to emphasize enhancing system robustness and security. This paper provides a comprehensive review of the latest advancements in FLLM, examining challenges from four critical perspectives: feasibility, robustness, security, and future directions. We present an exhaustive survey of existing studies on FLLM feasibility, introduce methods to enhance robustness in the face of resource, data, and task heterogeneity, and analyze novel risks associated with this integration, including privacy threats and security challenges. We also review the latest developments in defense mechanisms and explore promising future research directions, such as few-shot learning, machine unlearning, and IP protection. This survey highlights the pressing need for further research to enhance system robustness and security while addressing the unique challenges posed by the integration of FL and LLM.

CCS Concepts: • **Computing methodologies** → **Machine learning**.

Additional Key Words and Phrases: Federated learning, large language models, federated large language models, efficient training, security, heterogeneity

Authors' Contact Information: Wenhao Jiang, jwh_roy@nudt.edu.cn, National University of Defense Technology, Changsha, China; Yuchuan Luo, luoyuchuan09@nudt.edu.cn, National University of Defense Technology, Changsha, China; Guilin Deng, dengguilin@nudt.edu.cn, National University of Defense Technology, Changsha, China; Silong Chen, chensilong@nudt.edu.cn, National University of Defense Technology, Changsha, China; Xu Yang, yangxu16@nudt.edu.cn, National University of Defense Technology, Changsha, China; Shihong Wu, wushihong@nudt.edu.cn, National University of Defense Technology, Changsha, China; Xinwen Gao, gaoxinwen17@nudt.edu.cn, National University of Defense Technology, Changsha, China; Lin Liu, liulin16@nudt.edu.cn, National University of Defense Technology, Changsha, China; Shaojing Fu, fusj_nudt@163.com, National University of Defense Technology, Changsha, China.

Manuscript submitted to ACM

ACM Reference Format:
Wenhao Jiang, Yuchuan Luo, Guilin Deng, Silong Chen, Xu Yang, Shihong Wu, Xinwen Gao, Lin Liu, and Shaojing Fu. 2025. Federated Large Language Models: Feasibility, Robustness, Security and Future Directions. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 35 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Large language models (LLMs), represented by DeepSeek and ChatGPT, have demonstrated remarkable performance in intelligent question answering, logical reasoning, and natural language processing. This has led to their significant success in various fields, such as biomedicine [51, 63], legal consulting [145], and recommendation systems [55, 149], and has once again sparked a research boom in artificial intelligence technologies. However, as the scale of these models continues to expand, the demand of raw data for pre-trained models (also known as foundation models [12], FMs) is also increasing, leading to growing data anxiety. The current trend is that a large amount of high-quality private data is widely distributed among various data holders. Due to privacy concerns, regulation restrictions, or other reasons, these data cannot be publicly shared, resulting in the emergence of data silos [47]. Villalobos et al. [108] found that high-quality public data will be exhausted before 2026, which will not be able to support the further development of LLMs. Meanwhile, the traditional centralized large-scale collection of private data for training will inevitably violate user privacy.

To address the aforementioned challenges, the integration of LLMs with federated learning (FL) [73] emerges as a silver-bullet solution. FL is a distributed machine learning paradigm that enables multiple parties to collaboratively train a single model using their private data while preserving data privacy. This union, known as Federated Large Language Model (FLLM), has recently become a burgeoning research hotspot. Although this powerful combination can overcome data limitations, it also introduces multifaceted challenges, including parameter aggregation for feasibility, heterogeneity for robustness, and privacy and security concerns.

LLMs are a type of FMs, and several recent reviews have explored the prospects of combining FL with FMs [50, 52, 91, 114, 135, 159], but only a few have focused on FLLM [18, 35, 129]. Distinct from these works, we examine FLLM from a novel perspective, focusing on four critical dimensions: Feasibility, Robustness, Security, and Future Directions, with an emphasis on Robustness and Security. Notably, existing surveys, due to the limited pool of early FLLM papers, predominantly cite studies on FL or LLM rather than FLLM itself. In contrast, our work surveys the latest FLLM research, offering readers a more direct and up-to-date understanding of the current research trends. Table 1 provides a comparative overview of our work and some previous surveys.

To address the feasibility of FLLM, it is essential to explore efficient implementation methods. Training LLMs involves updating parameters at the trillion scale, which generates substantial communication and computational overheads that are unacceptable for resource-constrained federated participants. To tackle this issue, Federated Parameter-Efficient Fine-Tuning (FedPEFT) [71] has emerged as a promising solution. Based on extensive research, we categorize the fine-tuning methods for FLLM into four types: full-parameter fine-tuning, parameter-efficient fine-tuning, prompt tuning, and other specialized fine-tuning methods. Currently, numerous studies have addressed the feasibility of FLLM from an academic perspective, demonstrating the potential for fine-tuning LLMs on client devices.

Robustness corresponds to the heterogeneity of FLLM. Unlike centralized training, where data is aggregated in a single location, FLLM is trained on discretely distributed data across various clients, each equipped with distinct hardware and software environments. This setup readily gives rise to heterogeneity issues, which we categorize into

Table 1. Comparison of our work with the Previous Surveys[1]

Year	Work	Topic	Feasibility	Robustness	Security	Future Directions
2023	Chen et al. [18]	FLLM	✓	✗	✔	✗
	Yu et al. [135]	FL-FM	✓	✗	✗	✓
	Zhuang et al. [159]	FL-FM	✔	✗	✔	✓
2024	Ren et al. [91]	FL-FM	✓	✗	✔	✓
	Li et al. [50]	FL-FM	✓	✓	✔	✓
	Woisetschlager et al. [114]	FL-FM	✓	✗	✗	✔
	Li et al. [52]	FL-FM	✓	✗	✔	✓
	Yao et al. [129]	FLLM	✓	✓	✔	✓
	Hu et al. [35]	FLLM	✓	✗	✔	✔
2025	Our work	FLLM	✓	✓	✓	✓

[1] ✓ represents high correlation, ✔ represents few references to the FLLM paper, and ✗ represents no relevant content.

three types: resource heterogeneity, data heterogeneity, and task heterogeneity. (i) Resource heterogeneity refers to the differences in the computational and storage resources available to clients, which affects the training process and aggregation strategies. Unlike traditional FL, the large scale of parameters trained during FLLM means that resource-constrained clients may not be able to efficiently complete the fine-tuning tasks, resulting in slow training processes and even incomplete training. (ii) Data heterogeneity refers to the unequal distribution of data across clients, that is, the data are non-independent and identically distributed (Non-i.i.d.) [19]. Different clients may have different update directions, leading to drift in the global model update. Due to the complex structure and strong memory of the model resulting from its parameters, FLLM is more affected by data heterogeneity compared to traditional FL, leading to greater instability. (iii) Task heterogeneity indicates that different clients may have different types of tasks [6, 19, 21, 84, 126]. Given the strong semantic understanding capabilities of LLMs, they can be applied to a wide range of tasks, such as question-answering, classification, text generation, and translation. This diversity in tasks across clients leads to significantly different convergence directions for the models, creating obstacles to the robustness of FLLM. Currently, several studies have focused on addressing the heterogeneity issues in FLLM to enhance their robustness. However, more in-depth research is still needed to tackle the challenges posed by data and task heterogeneity.

Security corresponds to the privacy and security of FLLM. Integrating FL into LLMs may simultaneously introduce privacy threats from both LLMs and FL, such as data reconstruction and membership inference[18, 32]. Security challenges mainly refer to the threats posed by malicious attackers or curious participants exploiting vulnerabilities to impair system performance. In the traditional FL, these primarily include model poisoning attacks[117], data poisoning attacks[11], backdoor attacks[127], and adversarial attacks[31]. In FLLM, textual data is characterized by its discrete nature, where each word or character corresponds to a discrete index. For gradient leakage attacks, this discreteness makes it more challenging to directly reconstruct the original text from the gradient information. For poisoning attacks, the complexity of the model and the discreteness of the data require more sophisticated strategies for the attacker to achieve effective poisoning. For inference attacks, attackers need to infer certain features of the text by analyzing the model's outputs or the activations of intermediate layers. These attacks may either affect the model convergence speed or even prevent convergence, degrade model performance, or cause incorrect inference results on specific data, ultimately reducing the credibility of the model.

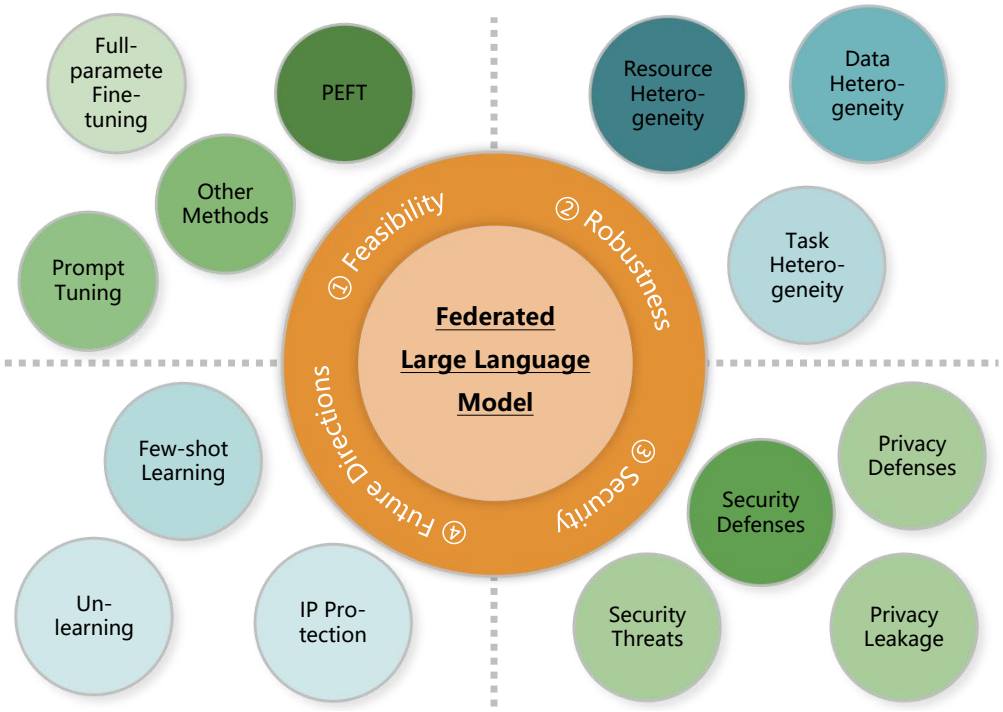

Fig. 1. The framework of this survey. Darker circles indicate a higher number of related studies.

The future direction primarily focuses on the issues that remain unexplored by FLLM. For example, making full use of scarce training data, protecting user rights, and protecting the intellectual property (IP) of models are additional requirements for FLLM to achieve long-term development. The significance of studying few-shot scenarios is substantial, as acquiring large amounts of labeled data is both time-consuming and costly. In training FLLM, insufficient labeled data can prevent the model from fully learning and understanding the patterns and features within the data, thereby affecting the model's training effectiveness and generalization ability. Moreover, the issue of catastrophic forgetting, which may arise as FLLM meets the need for clients to delete private training data, poses a significant barrier to its development. Furthermore, in terms of intellectual property protection, ensuring that the model is not illegally copied, redistributed, or misused has become a crucial challenge for FLLM.

In this survey, we focus primarily on recent papers concerning FLLM from four aspects: feasibility, robustness, security, and future directions. We present the framework of our survey in Figure 1, where the darker circles indicate a larger number of related studies. Unlike previous reviews, we place particular emphasis on the novel issues of heterogeneity and privacy and security arising from the integration of FL and LLMs, as well as the corresponding solutions, and have collected the latest FLLM-related papers. Our goal is to provide valuable references for researchers and practitioners interested in this interdisciplinary field. We systematically review the latest methods, highlight their contributions, and discuss how they address the inherent challenges of combining FL with LLM. Finally, we emphasize potential research directions that may emerge in the near future and summarize the entire text.

2 Feasibility of FLLM

FLLM essentially involves training LLMs using the methodology of FL. The conventional process of training an LLM comprises two primary stages: pre-training and fine-tuning. Currently, the number of parameters in LLMs has reached hundreds of billions and continues to increase. Companies such as Google, OpenAI, and Huawei have all proposed the development of models with trillions of parameters. For the majority of clients participating in FLLM, they lack the abundant resources of a server, making it impractical to jointly train LLMs from scratch. Therefore, FLLM predominantly focuses on the fine-tuning phase. Under the premise of preserving privacy, fine-tuning FLLM with vertical domain data represents the optimal approach for fine-tuning on private data for downstream tasks. However, in terms of computational costs, the substantial computational demands of backpropagation make it challenging for clients to fine-tune LLMs. Even with GPU clusters, training such models remains a formidable challenge. Regarding communication costs, since FLLM necessitates the sharing of model gradients or aggregated model parameters, the limited network bandwidth of clients and the extensive transmission of parameters can lead to communication bottlenecks between clients and the server.

To overcome the challenges posed by computational and communication costs and to realize the feasibility of FLLM, a series of Federated Parameter-Efficient Fine-Tuning (FedPEFT) methods have continuously emerged [71]. In this section, we categorize the training methods according to the modifications made during the training process of FLLM into full-parameter fine-tuning, parameter-efficient fine-tuning, prompt tuning, and other special techniques, and introduce each method respectively, as shown in Table 2. To clearly delineate the distinctions among various fine-tuning methods, we have indicated the trainable parameters in orange in Figure 2.

Table 2. Overview of feasibility in FLLM.

Type	Describe	Approaches
Full-parameter Fine-tuning	Fine tune all of the parameters	Fedlegal [145], FedRDMA [143]
Parameter-Efficient Fine-tuning (PEFT)	Fine tune part of the parameters, or freeze parameters of LLM and insert additional trainable modules.	FedCyBGD [112], BitFit [136], FeS [13], FedSelect [104], AdaFL [16], FlexLoRA [6], HETLORA [21], FwdLLM [120], LP-FL [39], DP-DyLoRA [119]
Prompt tuning	Fine-tunes prompts without altering the parameters of LLMs.	Promptfl [29], FedTPG [87], TCFL [150]
Other methods	Model compression, split learning, and zeroth-order optimization.	[116, 124, 134], [57, 152], [58, 66, 86]

2.1 Full-parameter Fine-tuning

Full-parameter fine-tuning refers to the process of continuing to train the parameters of a pre-trained LLM on a private dataset, which involves updating all parameters. During fine-tuning, the model architecture remains unchanged, and only the existing parameters are optimized to adapt to specific downstream tasks.

Full-parameter fine-tuning is the most straightforward method of parameter fine-tuning, where all model parameters are updated during the fine-tuning process. Zhang et al.[145] achieved the first FLLM in the legal domain by federated fine-tuning the RoBERTa-WWM pre-trained model released by HuggingFace on private information. However, the

communication and computational costs associated with this method are highly prominent issues. To address this, Zhang et al.[143] integrated RDMA technology into the FL framework to improve communication efficiency and robustness, thereby realizing the FedRDMA framework for FLLM fine-tuning based on an industrial FL framework. Compared with traditional TCP/IP-based FL systems, FedRDMA improves communication efficiency by 3.8 times. Nevertheless, this framework merely enhances communication efficiency through RDMA technology without resolving the significant computational costs, storage demands, and communication overheads associated with full-parameter fine-tuning. Wang et al.[112] proposed a full-parameter fine-tuning method called FedCyBGD to tackle these issues. This method uses cyclic block gradient descent to divide the model into multiple blocks, with each client responsible for updating one or several specific blocks cyclically, rather than the entire model, thereby reducing computational and storage costs. In addition, a model compression scheme was designed to lower communication costs, enabling full-parameter fine-tuning with lower resource consumption. However, this approach lacks sufficient convergence analysis and is limited by the small number of clients considered in its experiments. Due to the inability of full-parameter fine-tuning to fundamentally solve the problems of high computational costs, storage demands, and significant communication overheads, independent research on this method is relatively rare, and it is not the preferred approach for FLLM. Instead, it is often used as a reference and comparison for other methods [136].

The advantages and disadvantages of full-parameter fine-tuning are both quite evident. This fine-tuning approach is highly adaptable, allowing for fine-grained adjustments to the model's internal representations to better fit downstream tasks. When sufficient resources are available, Full-parameter fine-tuning typically yields the best performance, making it widely applicable across various model architectures and task types. However, it requires updating and transmitting a large number of parameters during each communication round, demanding high hardware requirements for the devices. Moreover, with limited data, overfitting is likely to occur, which in turn degrades the model's generalization ability.

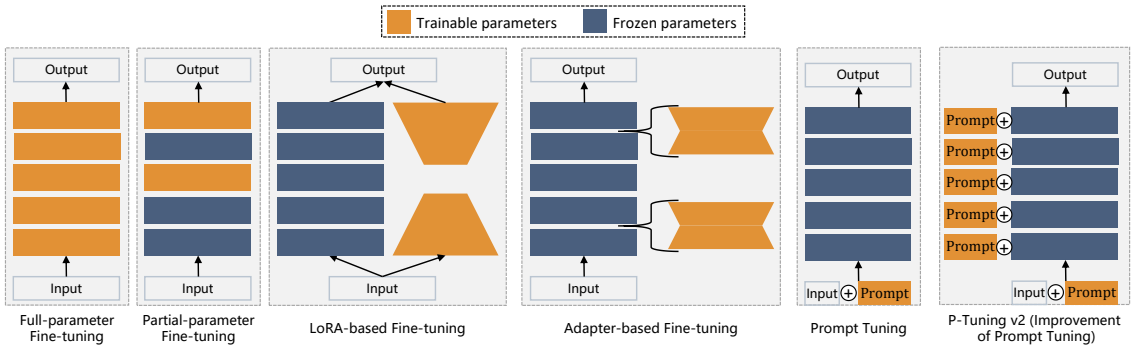

Fig. 2. The fine-tuning process of full-parameter fine-tuning, PEFT (including partial-parameter fine-tuning, LoRA, Adapter) and prompt tuning. The orange box denotes the trainable parameters.

2.2 Parameter-Efficient Fine-Tuning

Federated Parameter-Efficient Fine-Tuning (FedPEFT) [71] is a series of methods aimed at reducing the computational and communication overhead of federated fine-tuning. We categorize these methods into three types based on the part being updated: partial-parameter fine-tuning, LoRA-based fine-tuning and Adapter-based fine-tuning.

Partial-parameter fine-tuning methods improve the full-parameter fine-tuning methods, reducing computational, storage, and communication costs by decreasing the number of trained parameters. Zaken et al. [136] proposed BitFit, a sparse fine-tuning method that updates only the bias parameters. This method achieves comparable or even better performance than fine-tuning the entire model with significantly fewer parameters on small-scale to medium-sized datasets, as full-parameter fine-tuning is prone to overfitting. However, the theoretical explanation for BitFit is insufficient, and whether the selection of bias terms is optimal remains questionable. Sun et al. [103] also focused on reducing overhead by updating only the bias part of the globally shared model, achieving good results. Cai et al. [13] extended this idea by proposing the FeS framework, which updates only the bias parameters of the intermediate layers and freezes the weights of the lower layers, resulting in improved training efficiency while maintaining good model performance. Rishub Tamirisa et al. [104] were inspired by the Lottery Ticket Hypothesis to introduce a method for finding the optimal parameters for local fine-tuning while freezing the remaining parameters. They use gradient information to identify the parameters that change the least during training, which are considered suitable for freezing and carrying shared knowledge, while parameters with significant changes are considered the best choices for local fine-tuning.

Adapter [33] and LoRA [34] (a.k.a. Reparameterization) freeze the entire pre-trained LLM and introduce some trainable small modules at different positions of the model structure to achieve model fine-tuning for downstream tasks. These small trainable modules usually have low-rank characteristics, which can significantly reduce the number of training parameters and thus lower the computational, storage, and communication costs. Adapter is inserted into the model in a "serial" manner, while LoRA is inserted in a "parallel" manner. Therefore, it is generally believed that adapter will cause additional inference latency, while LoRA will not introduce new inference latency. Adapter-based fine-tuning inserts small adapter modules into each layer of the pre-trained model and fine-tunes only the parameters of these adapter modules to adapt to downstream tasks, as shown on the left side of Figure 3. LoRA (Low-Rank Adaptation) introduces low-rank decomposition matrices into specific layers of LLMs, as shown on the right side of Figure 3. For a weight matrix W, LoRA represents its update as $\Delta W = BA$, where B and A are low-rank matrices, and $r \ll d$ (where d is the dimension of the original matrix, and r is the dimension of the low-rank matrix). During training, the original weight matrix W is frozen, and only the low-rank matrices B and A are trained. Thus, during forward propagation, the model's output can be expressed as $h = Wx + BAx$.

In FLLM with adapters, after the client training is completed, the adapter parameters are sent to the server for aggregation, and then the aggregated adapter parameters are broadcast back to the clients for the next round of training. Cai et al. [16] proposed a FLLM scheme with dynamically configured adapters based on the adapter configuration method in [81], and provided a search method for the optimal adapter configuration, supplemented by cache activation technology to achieve efficient federated fine-tuning of pre-trained LLMs. Subsequently, they proposed the FedAdapter [15] scheme, which inserts adapters into different layers of the pre-trained model for training, adopts a dynamic configuration mechanism to adjust the depth and width of the adapters according to the training progress, and further optimizes the computational efficiency of the clients through activation caching technology. However, the dynamic configuration process of adapters in FedAdapter requires additional computational and communication overhead, and the configuration space of adapters (depth and width) may not cover all optimization solutions. FedOA [132] enhances the model's generalization ability in out-of-distribution scenarios by combining adapters with feature distance regularization techniques and the global model's invariant feature learning capability, effectively addressing the challenges posed by large-scale parameters and data heterogeneity in FLLM. However, its convergence speed and stability in practical applications may be affected by the client data distribution and communication frequency, and it is

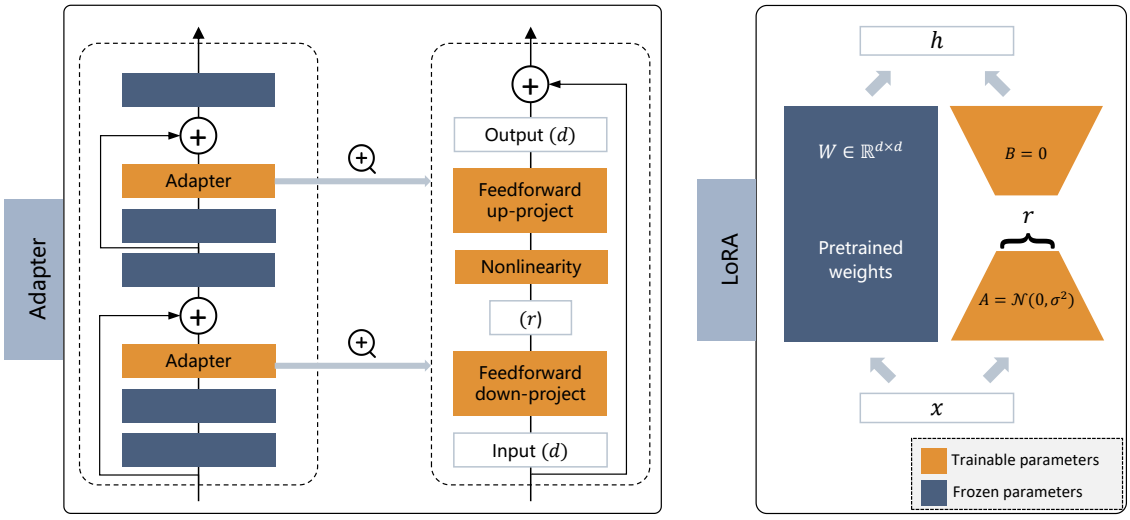

Fig. 3. The process of Adapter-based Fine-tuning and LoRA-based Fine-tuning.

sensitive to the selection of hyperparameters (such as regularization strength λ). Jia et al. [37] proposed the HeteroTune scheme, combining multi-branch cross-model aggregators to achieve efficient knowledge aggregation of models of different sizes on heterogeneous devices.

For FLLM with LoRA, different frameworks have different implementation processes. For example, in the FedLoRA framework [131], clients upload their trained A and B matrices to the server, which aggregates these matrices to generate global A and B matrices and broadcasts them back to the clients. Clients then use the global A and B matrices for further optimization in the next round of training. Using this LoRA fine-tuning method, Jiang et al. [39] implemented federated fine-tuning of pre-trained LLMs and found through experimental comparisons that this method can achieve performance similar to or even better than full-parameter fine-tuning. Bai et al. [6] designed the FlexLoRA method, which allows clients to dynamically adjust the rank of LoRA matrices based on local resources to change the number of trainable parameters. After aggregating the global LoRA, the server uses singular value decomposition to redistribute the parameters. Cho et al. [21], based on LoRA fine-tuning, employed techniques such as rank self-pruning and sparse weighted aggregation to further improve the convergence speed of the global model. Fang et al. [25] designed methods for identifying trainable weight importance and fast search algorithms to quickly search for the optimal low-rank adaptive matrices locally and further reduced storage requirements using quantization techniques. To further reduce memory overhead, researchers have integrated techniques such as forward differentiation with LoRA fine-tuning [79, 119, 120].

Adapter and LoRA methods are both representative techniques of PEFT and are suitable for fine-tuning LLMs in FL. Adapter inserts lightweight modules and is suitable for a variety of tasks (such as NLP, CV) and heterogeneous data distribution scenarios, with performance close to full model fine-tuning, but with higher communication overhead; LoRA, on the other hand, decomposes weight matrices into low-rank matrices and is particularly suitable for NLP tasks and homogeneous data distribution scenarios, with low communication overhead, but may have limited expressive power in complex tasks. Overall, adapter is more suitable for diverse tasks and heterogeneous data, while LoRA performs better in NLP tasks and scenarios with low communication demands.

2.3 Prompt tuning

Prompt tuning can fine-tune continuous prompts(a.k.a. soft prompts) without altering the parameters of the LLM [42], and is an emerging promising method for reducing costs and protecting privacy. The embedding layer, which is the initial component of the model, maps discrete words or tokens into a continuous vector space. These embedding vectors capture the semantics and contextual information of the words, forming the basis for the model's subsequent processing. The feasibility of prompt tuning is attributed to the fact that pre-trained language models (such as GPT, BERT) have acquired extensive linguistic knowledge and universal semantic representations through unsupervised learning on large-scale corpora. This universality provides the foundation for prompt tuning. Essentially, a prompt serves as a guiding signal, which directs the model to generate the desired output by incorporating specific text segments or structures into the input text. The core of prompt tuning lies in utilizing the model's embedding layer to implement trainable prompts, which are adjusted through optimization methods such as gradient descent to better adapt to specific tasks. During the process of prompt tuning, a set of prompt embedding vectors generated via the model's embedding layer is first initialized. Subsequently, these prompt embedding vectors are combined with task-specific input data to form the complete input. By optimizing the prompt embedding vectors using labeled data, the model can better adapt to specific tasks. In the inference stage, the optimized prompt embedding vectors are employed to generate the output for the task, thereby enhancing the model's performance on specific tasks. Since the number of parameters in the prompt embedding vectors is relatively small, updating these parameters is much more efficient than updating the entire model's parameters, making prompt tuning feasible on resource-constrained devices. As for FLLM, the classical approach of prompt tuning involves each client generating specific prompts for its local data and fine-tuning these prompts. Subsequently, the updated prompts are transmitted to the server for aggregation. The server then broadcasts the aggregated prompts back to the clients for the next round of training, as proposed in the referenced paper [29].

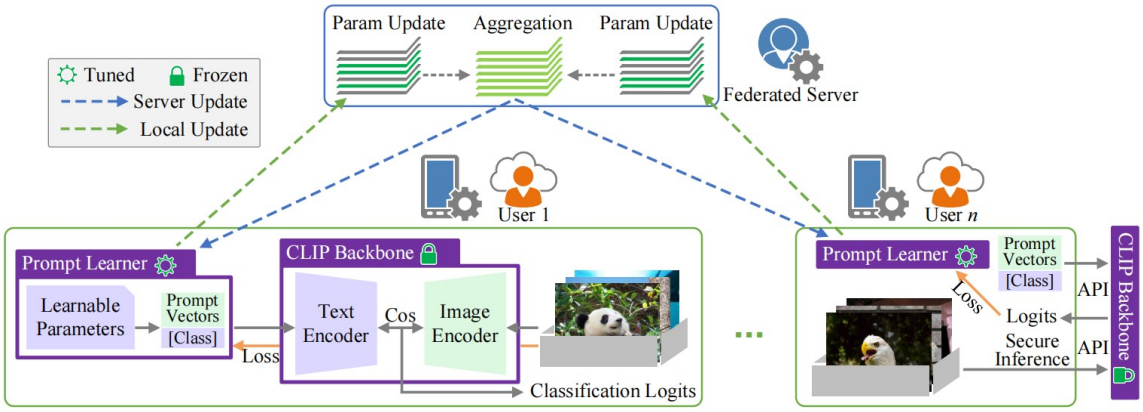

Fig. 4. The framework and workflow of PROMPTFL from [29]. Clients locally train a prompt learner with a small number of parameters, without altering the original LLM, and the server aggregates only the updates from the prompt learners.

The effectiveness of training prompts stems from the fact that LLMs are already quite intelligent. Pre-trained models (such as GPT, BERT, etc.) have already learned a vast amount of linguistic knowledge, but they require a clear "guidance" to complete specific tasks. The prompt serves as this "guidance." Training prompts only adjusts the prompt embeddings, not the parameters of the entire model, much like a teacher instructing a student on how to understand a question,

rather than reteaching the student all the knowledge. Guo et al. [29] generated a prompt learner on the client side, transforming federated model training into federated prompt training. Taking the pre-trained LLM CLIP [89] as an example, clients train soft prompts with a small amount of local data, send the updates of the prompt learner to the server for aggregation, and then update the local prompt learner based on the feedback from the server, as shown in Figure 4. This method of prompt tuning only requires training a small-scale prompt learner, without the need to fine-tune the LLM itself, thereby significantly reducing computational and communication costs. Qiu et al. [87] proposed the FedTPG scheme, which involves learning a unified prompt generation network (Prompt Generator) globally to convert task-related text inputs into context-aware prompt vectors. This approach transforms the training of LLMs into the training of the prompt generation network, not only enhancing the model's adaptability but also maintaining low communication costs, making it suitable for FLLM scenarios. Zhao et al. [150] proposed the TCFL scheme. This dual-prompt FL integrates visual and textual modalities to overcome the limitations of single-modality prompt tuning in FL, thereby improving data representation among nodes.

In addition, P-Tuning v2 [61] is an improved deep prompt tuning method that aims to address the limitations of traditional prompt tuning across models of varying sizes and tasks. By introducing trainable prompt embeddings at each layer of the model, it significantly enhances model performance, enabling it to match fine-tuning in a variety of natural language understanding (NLU) tasks.

The advantages of prompt tuning lie in its efficiency, privacy protection, as well as adaptability and flexibility. Given that the number of parameters in prompt embedding vectors is relatively small, updating these parameters is much more efficient than updating the entire model's parameters. This makes prompt tuning feasible on resource-constrained devices. In FLLM, the updates of prompt embedding vectors can protect the privacy of user data, as clients only need to share the updates of prompt embedding vectors, rather than the original data. Moreover, prompt embedding vectors are continuous in the embedding space and can be freely adjusted, without being restricted by natural language vocabulary. This flexibility enables prompts to better capture the complexity of tasks. Through careful design and optimization of prompts, prompt tuning can enable large language models to perform well in a variety of application scenarios.

2.4 Other Methods

In addition to parameter fine-tuning and prompt tuning, several other techniques have also contributed to the feasibility of FLLM. For instance, the introduction of model compression, split learning, and zeroth-order optimization has reduced the computational and communication overhead of FLLM, further facilitating its practical implementation.

Model compression minimizes the size of LLMs without compromising performance, thereby reducing the number of parameters that need to be trained [116, 124, 134]. This approach effectively reduces computational, memory, and communication costs. Researchers from Google, including Yang et al. [124], proposed the Online Model Compression (OMC) technique for lightweight operations, which compresses model parameters. Parameters are only decompressed and released into memory when they are involved in computations, and the model is shared and stored in a compressed format, thus reducing storage and communication costs. Model parameter quantization techniques reduce the storage and computational resources required by models by lowering the representation precision of model parameters, thereby reducing resource costs[25, 124, 156]. Yang et al. [124] studied the impact of parameter quantization on model performance from three different dimensions: full parameter quantization, weight-only quantization, and partial variable quantization, demonstrating the feasibility of quantization techniques.

Split learning approaches the problem from the perspective of model architecture by partitioning the LLM into several sub-models of varying sizes, which are then distributed across the server and clients. This ensures that the

primary computational load during training remains on the server. This method significantly reduces the resource demands on client devices in FLLM and has been applied in FLLM [57, 152].

Zeroth-order optimization techniques [58, 66, 86] consider the training process, achieving the feasibility of FLLM by adjusting the backpropagation process during training, optimizing model parameters without directly computing gradients, greatly reducing time costs. Noting that the resource demands of neural network training are significantly higher than those of inference, Xu et al. [120] proposed an efficient federated fine-tuning of LLMs based on perturbation inference without the backpropagation process. This method uses forward inference to determine the correctness of perturbations and combines specially designed perturbation discrimination methods to quickly eliminate perturbations that are almost orthogonal to the true gradient, achieving efficient training. Despite the efficient perturbation discrimination method, forward inference still needs to be executed multiple times to obtain an unbiased estimate of the true gradient. Panchal et al. [79] proposed estimating gradients by computing Jacobian-Vector Products based on random perturbations of weights during the forward pass. This allows gradient estimation to be completed with only two forward inferences, further improving training efficiency.

The feasibility of FLLM remains a recent research hotspot. At its core, the issue is to address the significant overhead associated with FLLM. The solutions we discussed have basically achieved the integration of FL and LLM in academia. Currently, FedPEFT methods are evolving towards lower computational, storage, and communication costs, enhancing the efficiency of FLLM on the basis of achieving feasibility.

3 Robustness of FLLM

Traditional FL already faces challenges of resource heterogeneity[46] and data heterogeneity[76] due to the distributed nature of training data across numerous clients with varying label distributions, sample sizes, and diverse device and network environments. Owing to the substantial parameter scale of LLMs, FLLM significantly exacerbates the issue of resource heterogeneity. Edge nodes with limited resources may not be able to participate in training normally due to computational and communication challenges. LLMs are highly complex and more sensitive to the quality and quantity of data, thereby exacerbating the impact of data heterogeneity. Moreover, FLLM introduces a novel issue: task heterogeneity. [6]. This arises because LLMs possess strong semantic understanding capabilities and can be applied to a wide variety of tasks, leading to different clients having distinct tasks such as question-answering, classification, text generation, and translation. The heterogeneous tasks on clients result in significantly different model convergence directions, which is markedly different from traditional FL. Heterogeneity issues are significant barriers to the practical deployment of FLLM, profoundly affecting their robustness. We have conducted a survey of recent works on resource heterogeneity, data heterogeneity, and task heterogeneity in FLLM, as shown in Table 3.

3.1 Resource Heterogeneity

In reality, FLLM systems are highly likely to be composed of devices with heterogeneous resources, featuring diverse software and hardware environments. This leads to differences among clients in terms of computing platforms, computational capabilities, storage space, communication bandwidth, and training efficiency, thereby affecting the training process and aggregation strategies [88]. FLLM is also subject to the "bucket effect," where efficiency is determined by the least capable participant, resulting in the waste of resources from other participants. Given the widespread existence of resource heterogeneity, existing research on the heterogeneity of FLLM mainly focuses on this issue [5, 6, 21, 96, 98, 101, 102, 105, 106, 122, 148]. In these studies, most methods are based on LoRA and adapter to address resource heterogeneity.

Table 3. Overview of robustness in FLLM.

Type	Describe	Approaches
Resource Heterogeneity	Diverse computational resources impact the aggregation efficiency of FLLM.	FedLoRA [131], FlexLoRA [6], Fed-piLot [148], HETLORA [21], FedRA [101], FedSpaLLM [5]
Data Heterogeneity	Unbalanced data distribution leads to global model drift.	FDLoRA [85], FedPipe [25], PFIT [38], SPRY [79], FwdLLM [74]
Task Heterogeneity	The demand for multitasking makes it difficult for FLLM to converge stably.	M2FEDSA [146], FedDPA [126], FedBone [20], FedDAT [19], FL-TAC [84]

The PEFT methods mentioned above not only contribute to the feasibility of FLLM, but also enhance their robustness. The FedLoRA framework [131] mentioned previously achieves parameter-efficient fine-tuning of large heterogeneous models by inserting a small LoRA on each client and aggregates LoRA on the server side to enable knowledge sharing among clients. Zhang et al. [148] proposed the Fed-piLot scheme, which, based on the observation that training different LoRA layers results in different memory consumption and that different layers contribute differently to model performance, formulated the allocation of LoRA as a knapsack optimization problem. They designed a value function based on local-global information gain score (IG-Score) to optimize the allocation of LoRA under client memory constraints. Cho et al. [21] first discussed that in resource-heterogeneous scenarios, the redistribution aggregation method using LoRA faces problems of overfitting and slow convergence, and thus proposed the HETLORA scheme. By applying local rank self-pruning and sparse weighted aggregation on the server, it combines the advantages of high- and low-rank LoRAs, achieving better convergence speed and final performance compared to homogeneous LoRA. Su et al. [101] proposed the FedRA scheme, which randomly generates an allocation matrix in each communication round to determine which layers of the model each client is responsible for updating. Resource-constrained clients only need to handle a small number of layers assigned to them and fine-tune through adapters, thereby reducing computational overhead. The server aggregates the adapter updates from clients to the corresponding layers of the global model according to the allocation matrix. This strategy not only fully utilizes the computational resources of different clients but also supports extremely heterogeneous scenarios where no client can fine-tune the entire model.

Furthermore, Bai et al. [6] proposed the FlexLoRA scheme, which allows clients to dynamically adjust local LoRA ranks based on their own resources, as shown in Figure 5. Unlike previous methods where the server aggregated matrices A and B as $B_g = (\sum_{i=1}^{m} n^i B_l^i)/(\sum_{i=1}^{m} n^i)$, $A_g = (\sum_{i=1}^{m} n^i A_l^i)/(\sum_{i=1}^{m} n^i)$, where m is the number of FL clients, n_i is the size of the i-th client's local training dataset, B_g, A_g are the global LoRA decomposed matrices, and B_l^i, A_l^i are the local LoRA decomposed matrices of i-th client, FlexLoRA aggregates the full-size LoRA $W_g = (\sum_{i=1}^{m} n^i W_l^i)/(\sum_{i=1}^{m} n^i) = (\sum_{i=1}^{m} n^i B_l^i A_l^i)/(\sum_{i=1}^{m} n^i)$, where W_l^i is the full-size LoRA of the i-th client. Clients with abundant resources can use higher LoRA ranks, thereby contributing more general, task-agnostic knowledge. This dynamic adjustment mechanism avoids the problem in traditional FL where the resources of other clients cannot be fully utilized due to the limitations of the client with the least resources.

In addition, there are also methods based on model pruning[5] and split learning[96, 106, 122] that can address resource heterogeneity. Bai et al. [5] first proposed the pruning scheme FedSpaLLM for FLLM, which allows clients to prune the model locally based on private data while considering system resource heterogeneity and maintaining

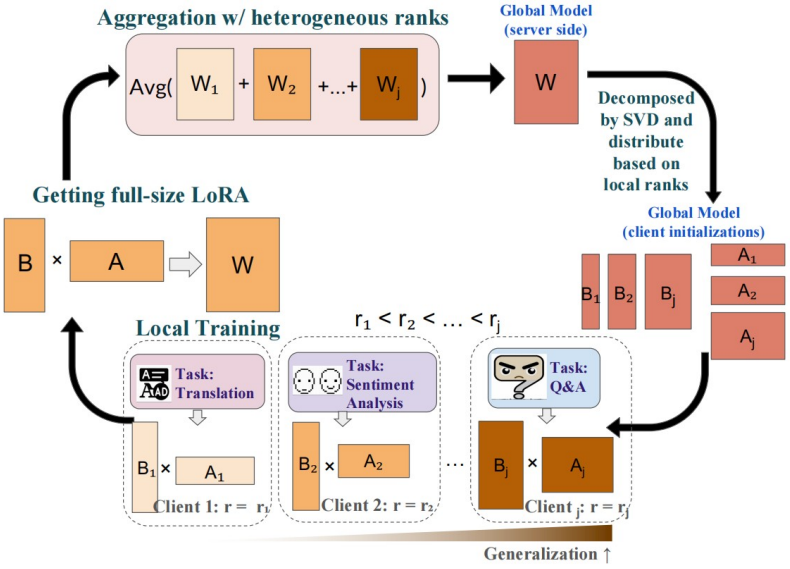

Fig. 5. FlexLoRA workflow from [6]. The server aggregates the full-size LoRA after multiplication, not the individual matrices A and B. The global full-size LoRA is then decomposed into smaller matrices of varying ranks via SVD and allocated sequentially based on client resources.

competitive communication efficiency. The server randomly samples a portion of the model layers for each client in each communication round, with the sampling quantity proportional to the client's computational resources. This proportional allocation ensures that each client can effectively participate in model training within its capabilities. Split learning, on the other hand, reduces the processing load on clients by dividing the complete model into smaller submodels. Clients can choose to run the initial layers of the model based on their computational resources, while the server is responsible for running the remaining layers. This division allows resource-constrained clients to handle only the lightweight parts of the model, leaving the computationally intensive tasks to the server.

The resource capacity of clients determines the size of the model they can train, which may lead to model heterogeneity. Samiul Alam et al. [1] proposed a model-heterogeneous training scheme based on partial training called FedRolex. Through a rolling sub-model extraction scheme, it enables the parameters of the global server model to be trained uniformly, thereby mitigating the client drift problem caused by model heterogeneity.

3.2 Data Heterogeneity

Data heterogeneity, that is, the imbalance in data distribution across different clients, unequal data volumes, and the non-independent and identically distributed (Non-i.i.d.) nature of the data [19], can lead to drift in local model updates during training. This means that the update directions of models on different clients are significantly different. As a result, the global model obtained by aggregating parameters from various client models on the server is unstable and usually not the optimal model [46, 76]. LLMs typically require substantial amounts of data for adequate training. However, in FL, data is distributed across multiple clients and is highly heterogeneous, making it difficult for the model to learn globally effective features from local data. Moreover, clients may only be exposed to specific patterns in their local data, leading to overfitting of local models and a decrease in the generalization ability of the global model.

Similar to resource heterogeneity, most current methods for addressing data heterogeneity are based on LoRA and Adapter. Qi et al. [85] proposed the FDLoRA scheme with a dual-module configuration. Each client is equipped with two LoRA modules, one for capturing local personalized knowledge and the other for global knowledge. Only the parameters of the global LoRA module are shared during parameter sharing. Clients then adaptively fuse the parameters of the dual LoRA modules based on the aggregated parameters from the server, achieving good performance, especially when the degree of data heterogeneity among clients is severe. Fang et al. [25] proposed the FedPipe scheme, whose core idea is to configure different LoRA matrices for each client. FedPipe first identifies the weights that contribute most to model training. By analyzing the importance of these weights, it selects the weights suitable for fine-tuning on edge devices. For each selected weight, FedPipe configures a LoRA that is trained on local data, dynamically adjusting the LoRA's parameters (such as batch size and decomposition rank) according to the computational and storage resource constraints of each edge device. This method ensures that each edge server can efficiently fine-tune the model within its resource limitations. Jiang et al. [38] proposed the PFIT method, which uses reinforcement learning to fine-tune local LLMs. Each client adjusts model parameters using different reward models based on the characteristics and needs of its local data. Clients use a global adapter to fine-tune the global model. The role of the global adapter is to enable the model to adapt to the characteristics of global tasks, but its parameter updates are fed back to the server for updating the global model. Meanwhile, clients train local LoRA on local data to achieve personalized adjustments. The parameters of these LoRAs are updated locally and are not uploaded to the server, thereby eliminating the heterogeneity of client data.

In addition, there are also some schemes that consider the training process and propose innovative methods. Panchal et al. [79] proposed SPRY, which uses Forward-mode Auto-Differentiation to fine-tune LLMs, achieving low memory usage, high accuracy, and fast convergence. When the data among clients are homogeneous, the global gradient aggregated on the server side by SPRY is an unbiased estimator of the true global gradient. Although heterogeneity increases the bias of the estimator, it remains usable. Mei et al. [74] addressed the challenges posed by data heterogeneity by employing a Mixture-of-Experts (MoE) model. The proposed FedMoE constructs an optimal sub-MoE model for each client and feeds knowledge back into the global MoE to improve efficiency in data heterogeneity environments.

In the FLLM environment, data heterogeneity is a key challenge. However, most of the current methods for addressing data heterogeneity focus on model architecture design, with fewer methods employing other techniques such as model distillation. We believe that more universal and efficient FLLM solutions for data heterogeneity will emerge soon.

3.3 Task Heterogeneity

Task heterogeneity refers to the fact that different clients may have different types of tasks. In real-life FLLM scenarios, computational and storage resources are very limited. Training separate models for each task would significantly increase the computational and storage burden, potentially exceeding the resource limitations of the devices. By sharing a single model, multiple tasks can be handled simultaneously within limited resources, improving resource utilization. For example, a LLM can handle tasks such as text classification, sentiment analysis, and machine translation simultaneously, without the need to train and deploy separate models for each task. Multi-task learning can enhance the model's generalization ability because it forces the model to learn more generic feature representations instead of overfitting to the data of a specific task. This generalization ability enables the model to perform better when facing new tasks. These advantages make multi-task learning a more practical and efficient choice in resource-constrained and privacy-sensitive environments. However, multi-task learning is prone to model forgetting issues [62], and conflicts in optimization objectives as well as differences in model structures pose significant challenges to the stable convergence of the global model.

Currently, research on task heterogeneity in FLLM is relatively limited, and the focus is on expanding model scale by setting up multiple adapters. Zhang et al. [146] proposed the M2FEDSA framework, which combines split learning and multi-modal FL. It introduces a dual adaptive fine-tuning strategy by adding task adapters in the high-level encoder on the main server and modality adapters in the low-level encoder on the client side to enhance the model's adaptability to different tasks and modalities. It also employs a dual knowledge transfer strategy to pass multi-modal knowledge to single-modal features at the feature and decision levels, further improving model performance. Yang et al. [126] proposed the FedDPA framework, which combines global and local adapters to learn general knowledge across different distributions and provide personalized services for each client. Additionally, FedDPA introduces an instance-based dynamic weighting mechanism that dynamically integrates global and local adapters during inference to achieve effective test-time personalization. Chen et al. [20] proposed the FedBone framework, which splits the model into a general model deployed on the cloud and a task-specific model deployed on the client side. The cloud's powerful computing capabilities handle general feature extraction, while the client is only responsible for lightweight data embedding and task output. FedBone also introduces the GPAggregation method, which calculates the attention values of task gradients and historical aggregated gradients and performs projection operations to eliminate conflicts between gradients of different tasks, enhancing the model's generalization ability. It designs a task adaptation module using deformable convolution and self-attention mechanisms to further enhance the model's adaptability to different tasks. Chen et al. [19] proposed the FedDAT scheme, which effectively addresses task heterogeneity through a dual-adapter teacher module and mutual knowledge distillation strategy, achieving distributed fine-tuning of the base model while maintaining communication efficiency. Experimental results show that FedDAT significantly outperforms existing centralized PEFT methods in multiple multi-modal FL benchmarks, demonstrating better convergence speed and scalability. Ping et al. [84] proposed a FLLM training method called FL-TAC, which trains a low-level adapter for each individual task on the client side and then clusters similar adapter groups on the server side for task-specific aggregation.

The heterogeneity issues of FLLM are primarily manifested in resource heterogeneity, data heterogeneity, and task heterogeneity. The ability to address these heterogeneity issues in FLLM is crucial for their robustness and has become one of the current research hotspots. Presently, most research focuses on the resource heterogeneity of FLLM, while further in-depth studies are needed for data and task heterogeneity.

4 Security of FLLM

The primary motivation for combining LLMs with FL is to enable LLM training that collects data from various parties while protecting user privacy. Therefore, privacy protection is the most important feature of FLLM. However, due to the emerging nature of FLLM, current research has focused excessively on how to achieve efficient fine-tuning of FLLM, with insufficient analysis of the system's privacy and security. Both FL and LLM have their own privacy and security threats. Whether the combination of the two will lead to an accumulation of all risks or mitigate some attacks has not yet been clearly summarized in any review. This section analyzes whether the attacks that originally existed separately in FL and LLM are still effective for FLLM and surveys the new risks generated by the combination of FL and LLM. After conducting an extensive search, we found that research on attacks targeting FLLM remains limited. Correspondingly, studies on defense mechanisms for FLLM are also scarce, posing significant challenges to its security.

In FL systems, although the server cannot access the raw data of clients, it may still be able to infer client privacy through the collection of model parameters or gradients. For example, membership inference attacks [109] and input reconstruction attacks [151] are potential threats. Malicious clients, aiming to control the direction of global model training or to prevent model convergence, may launch data poisoning attacks [111, 118, 147] and model poisoning

attacks [4, 10, 28, 110], as shown in Figure 6. LLMs are trained on vast amounts of public data, which may include private information such as email addresses and phone numbers. As high-quality public data becomes scarce, training LLMs on private data is becoming more common, potentially exposing private data to privacy breaches. Prominent methods for privacy theft include membership inference and reconstruction attacks. Additionally, during training, LLMs not only learn the underlying logic of language but also potentially acquire knowledge that contradicts human values, posing the risk of generating harmful content. Jailbreaking attacks are a primary means that may trigger this risk. Furthermore, the combination of FL and LLMs introduces new privacy and security threats. Training data for LLMs can inadvertently be reflected in generated content, potentially revealing sensitive personal information such as medical records and bank account numbers. This allows other clients in the FL system to potentially steal user privacy. Due to the deep transformer architecture and multi-stage training process of LLMs, poisoning attacks are more likely to succeed and are harder to detect in FLLM training. The high reliability of FLLM is a key factor for their successful implementation. We provide an overview of privacy and security attacks and defenses of FLLM in Tables 4 and 5, respectively.

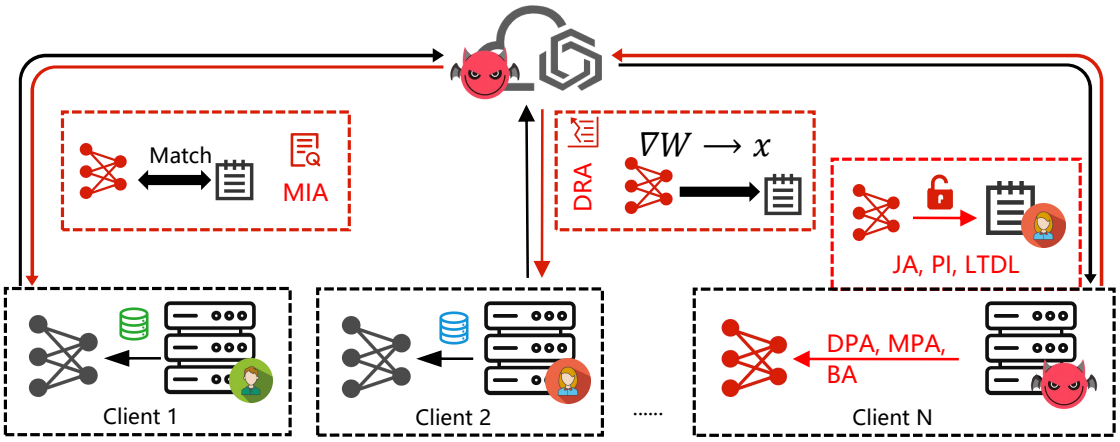

Fig. 6. Privacy and security threats faced by FLLM system. Honest-but-curious server steals client privacy through MIA and DRA. Malicious clients steal other users' privacy through JA, PI, and LTDL, and also damage or manipulate the aggregation of the global model through DPA, MPA, and BA.

4.1 Privacy Leakage and Defenses

4.1.1 Membership Inference Attack. Membership Inference Attack (MIA) aims to determine whether a target sample exists in a specific participant's training set [97]. Bai et al. [7] categorized the implementation methods of membership inference attacks in FL into update-based and trend-based approaches. One update-based method involves using model gradients as attack feature vectors [30, 77] or comparing the differences in gradients across different rounds [45, 75] to infer membership status. However, given that FLLM training requires datasets on the scale of billions, the presence or absence of a single training data point may have a negligible impact on gradient changes, making it difficult to determine whether it has been trained on. This has been confirmed in the research by Duan et al. [22]. They conducted a large-scale evaluation of a series of language models (ranging from 160M to 12B parameters) and found that in most

Table 4. Overview of privacy in FLLM.

Type		Describe	Approaches
Attacks	Membership Inference Attack	Determine whether a specific data point was used.	[109]
	Data Reconstruction Attack	Recover original data from model outputs or parameters.	[80], [151]
	Jailbreaking Attack	Exploit vulnerabilities in system constraints to gain unauthorized access.	[49]
	Prompt Injection	Manipulate input prompts to elicit malicious outputs.	-
	Long-Tailed Data Leakage	Infer sensitive information from underrepresented classes in imbalanced datasets.	-
Defenses	Multi-layered defense	Adapt multi-layered defense mechanism, such as weighted aggregation, geometric median, model pruning, and noise addition to resist attacks.	FedSecurity [32]
	Parameter quantization	Integrating quantization and LoRA to exchange only partial model parameters during training.	FedLPP [156]
	Differential privacy	Combine differential privacy with adapter mechanisms.	FedPA [139]
	Local fine-tuning	Fine-tuning LLMs directly on client devices with private data.	Titanic [99]

cases, the performance of membership inference attacks only slightly outperformed random guessing, which is in stark contrast to previous findings in traditional machine learning models. This inefficiency is primarily attributed to two characteristics of LLMs: first, the use of massive amounts of data for training, which makes it difficult for the model to overfit the training data; and second, the training process typically involves only a little over one epoch, resulting in strong model generalization and difficulty in distinguishing between training and non-training data.

Another update-based method infers membership status through structure modifying [78, 83]. A malicious server can meticulously design the model structure, embedding malicious parameters that activate when the target member participates in training, thereby leaking membership information. Although this method places high demands on how the server designs the model structure, similar attacks have been realized in FLLM [109]. Minh N. Vu et al. [109] proposed two methods for a malicious server to launch active membership inference attacks in the context of FLLM. The first involves setting the first two layers of the model as fully connected layers and carefully designing the weights and bias parameters based on the target sentence or token. The presence of zeros in the output of these layers determines whether the target exists in the input. The second focuses on the self-attention layer, similarly designing the parameters of the Q, K, and V matrices of the self-attention layer based on the target token and analyzing the output values to determine the target member's attributes. However, the paper assumes that the attacker can fully control the global model and manipulate its architecture, which may not entirely hold in practical applications.

Trend-based methods leverage the change trajectories of model outputs or parameters across multiple rounds to infer membership information [137, 142], requiring only the observation of metric changes in each round with minimal

time overhead. This method is simple and efficient to implement and could potentially be realized in FLLM, although such instances have not yet appeared.

4.1.2 Data Reconstruction Attack. Data Reconstruction Attack (DRA) is a type of attack that reconstructs original data from gradients or other information. Yang et al. [123] categorized gradient leakage attacks in FL into two major types: optimization-based attacks and analysis-based attacks, and proposed a new generation-based GLA paradigm, demonstrating its advantages in terms of data reconstruction performance and efficiency. Optimization-based attacks generate initial data randomly and optimize the data based on gradients to make it close to the private training data, thereby achieving data reconstruction. For example, Mu et al. [157] studied a gradient-based reconstruction attack algorithm, mainly using deep learning techniques and algorithms to analyze the gradient information in FL models to recover the original training image data. They proposed an algorithm called deep leakage from gradient (DLG), which can recover the original image by generating virtual image labels and calculating virtual gradients to match the real gradients, without access to the original dataset. This optimization-based method, relying solely on gradient changes, remains applicable to the fine-tuning methods of LLM parameter updates. In contrast, analysis-based attacks solve a system of linear equations to obtain the original private data, which is more accurate but only suitable for smaller models [26, 82, 155]. For LLMs with a huge number of parameters, this type of attack is more difficult to implement.

In the context of FLLM training, adversaries attempt to recover users' input text, and the discrete nature of this input increases the difficulty of data reconstruction attacks. Zheng et al. [151] explored the security issues of implementing vertical FL for LLMs in a white-box scenario, pointing out that attackers can easily and at low cost reconstruct users' input text from the intermediate embedding layers, and discussed several possible solutions to enhance the privacy protection of vertical federated LLMs. Petrov et al. [80] proposed DAGER, a gradient inversion attack algorithm targeting FLLM, capable of accurately recovering entire input text batches from shared gradients. DAGER leverages the low-rank structure of gradients in self-attention layers and the discrete nature of token embeddings, employing exhaustive heuristic search and greedy methods to precisely recover batches for both encoder-based and decoder-based architectures. Lu et al. [65] proposed APRIL, a novel attack method that analyzes the gradient leakage risks of self-attention mechanisms and demonstrates that attackers can use the shared gradient updates of models to recover private training data within the FL framework. The paper particularly observed that learnable positional embeddings are a weak link in the privacy protection of Transformer models. Fowl et al. [27] proposed the DECEPTICONS attack method, which deploys malicious parameter vectors in FL to leak users' private text data, utilizing the characteristics of the Transformer architecture and token embeddings to extract token and positional embeddings separately to recover high-fidelity text, even in the face of small batches, multiple users, and long sequences. Rashid et al. [90] identified which training rounds included the participation of victims using the victim round identification method and proposed the maximizing data memorization method based on selective weights optimization and weights transformation learning to further enhance the model's memorization of sensitive data, significantly increasing the success rate of private data reconstruction (up to 71%).

4.1.3 Jailbreaking Attack. Jailbreaking Attacks (JA) refer to bypassing or breaking through the security and censorship functions of a model to perform unauthorized operations or output non-compliant content [49, 121], which has become one of the unique and mainstream attacks against LLMs. In the FLLM scenario, the training process involves multiple participants, each of whom can be a potential target for jailbreaking attacks. Attackers may inject malicious data into the training set or craft malicious prompts to steal other users' privacy from the global model. These attacks typically involve designing clever prompts to induce the model to exceed its preset limitations while performing tasks, thereby

achieving the attacker's goal. Usually, attackers launch jailbreaking attacks on LLMs through "jailbreaking prompts." Initially, "jailbreaking prompts" were mainly designed manually, which had limitations in terms of readability and fluency, and were later improved in subsequent work [40]. As jailbreaking attacks continue to evolve, diffusion models have been employed for generating "jailbreaking prompts" [113], significantly increasing the success rate of jailbreaking attacks while optimizing the fluency and diversity of the prompts. Current defense measures require security-oriented training and adversarial training at the model level to enhance the model's resistance to attacks [49]. Although various robust FL mechanisms, such as robust aggregation schemes [48], can protect training from malicious updates, their effectiveness against emerging jailbreaking threats remains to be explored.

4.1.4 Prompt Injection. Prompt Injection (PI) is a technique that uses malicious instructions as part of the input prompt to manipulate the output of a language model. In the FLLM scenario, where the models are exposed to a large number of unfamiliar users, prompt injection attacks exhibit characteristics of being difficult to predict and defend against, and having a strong immediacy. It is similar to SQL injection attacks in database security, where carefully crafted inputs bypass the model's normal processing procedures to achieve unauthorized data access, execute malicious code, or produce harmful outputs. Prompt injection can be divided into two forms: direct injection and indirect injection. The former involves directly adding malicious instructions to user input, while the latter hides malicious instructions in documents that may be retrieved or ingested by the model [36]. This type of attack primarily affects the integrity and security of applications based on LLMs, potentially leading to unauthorized data access, execution of malicious code, or generation of harmful outputs. General defense methods involve strengthening input validation and filtering at the application level to prevent untrusted user input from being directly passed to the LLM.

4.1.5 Long-Tailed Data Leakage. Long-tailed data leakage (LTDL) refers to the over-memorization of rare data by a minority of participants. This characteristic is particularly dangerous in FLLM because the local data of participants often contains sensitive information (such as medical records and financial transactions), and the global model may inadvertently leak these details through parameter aggregation. On the one hand, due to the large number of parameters and multi-layer attention mechanisms in LLMs, they possess extremely strong data representation capabilities, but also face the risk of over-memorizing long-tailed data [23]. Studies have shown that even when facing extremely few samples (such as rare case data from a certain participant), the model may accurately memorize data details through subtle changes in gradient updates. For example, Liu et al. [60] verified a variety of adversarial attack techniques targeting LLMs and found that high attack success rates can still be achieved with a small number of attack samples, especially on LLaMA-7B, where the ASR of 8-shot attacks always remains above 50%.

On the other hand, the training method of FLLM itself also exacerbates the severity of this problem. Taking a trillion-parameter model as an example, a single gradient update requires the transmission of hundreds of GB of parameters, and the frequent global aggregation in FL leads to exponential growth in communication bandwidth and computational resource consumption [128]. For instance, training a model of the scale of GPT-3 requires thousands of GPUs to work in parallel for several months, and the dispersed participants in FL may be forced to reduce the frequency of aggregation due to hardware heterogeneity (such as insufficient computing power of edge devices), which exacerbates the overfitting of local models to long-tailed data. Ma et al. [68] pointed out that the Non-i.i.d. data distribution will intensify the training directionality problem of the model, and the low-frequency update strategy is difficult to eliminate the parameter bias in the local model during global aggregation.

Research on the long-tailed privacy leakage risks in FLLM is currently still in its infancy. However, some foundational work on FL or LLMs can offer valuable insights. For instance, adding noise (such as Gaussian or Laplacian noise) to

the gradients or parameters of FLLM can mitigate the risk of data memorization. Batool et al. [9] proposed a VANETs FL framework that implements a lightweight privacy budget allocation strategy through differential privacy design, optimizing model aggregation efficiency while ensuring privacy security. However, this solution experiences a more significant accuracy loss in Non-i.i.d. data scenarios. Additionally, Mao et al. [72] suggested reducing the number of trainable parameters through LoRA, splitting LLMs into shared and private layers, and aggregating only the shared parameters. This approach can reduce parameter scale and memory effects to some extent but still faces communication efficiency issues in models with hundreds of billions of parameters.

4.1.6 Defenses. The privacy protection of FLLM faces numerous challenges, including data leakage risks and model protection. Recently, researchers have proposed a variety of privacy protection schemes, ranging from quantization techniques and differential privacy to distributed training paradigms, aiming to balance the relationship between privacy protection and model performance.

In the FedSecurity framework proposed by Han et al. [32], FedDefender, as a key component, is specifically designed for the defense mechanisms of FLLM to counter various attacks. FedDefender implements defensive measures at different stages of FL training, including the "pre-aggregation," "during-aggregation," and "post-aggregation" phases. Before aggregating client models, FedDefender can score local models to identify potentially malicious ones and reweight them to mitigate the impact of malicious models. For example, the Krum algorithm tolerates a certain number of Byzantine clients by selecting the single most likely benign model as the global model. During the aggregation process, FedDefender modifies the aggregation function to make it more robust against potential malicious client models. For instance, Robust Federated Aggregation calculates the geometric median of client models as the aggregated model instead of simply averaging them. After aggregation, FedDefender can directly modify the global model by clipping or adding noise to protect it from potential attackers. For example, Clipping-based Robust FL clips the global model after each aggregation to limit the model's norm. Through these multi-layered defense mechanisms, FedDefender can flexibly respond to different types of attacks, including data poisoning, model poisoning, and data reconstruction attacks.

Zhu et al. [156] proposed the FedLPP framework, which combines quantization techniques and LoRA to protect both data and model privacy in FL. FedLPP distributes quantized rather than complete model parameters during training, preventing clients from obtaining the full model on the server and effectively protecting the privacy of the global model. By updating only a small portion of the model's parameters, FedLPP further reduces communication overhead and limits clients' access to the global model's details.

Zhang et al. [139] suggested that LLM privacy protection can be achieved through FL frameworks and personalized adapter mechanisms. Each client learns a lightweight personalized adapter using its private data, which collaborates with the pre-trained base model to provide efficient and fine-grained services for recommendation systems. Throughout the process, users' private data remains on local devices and is not shared with the server, ensuring data privacy. The method further enhances privacy protection with differential privacy techniques, such as adding noise to model parameters when clients upload them to prevent the server from inferring users' original data. This data-localized privacy protection mechanism not only safeguards users' privacy but also allows models to integrate shared knowledge without sharing sensitive information while retaining each user's personalized preferences.

Su et al. [99] proposed the Titanic scheme, which deploys the fine-tuning process of LLMs directly on client devices holding private data. This approach ensures that private data always remains on local devices and is not sent to the cloud or other centralized servers, thereby maximizing data privacy protection. However, this method is impractical for resource-constrained clients. To address the challenge of client resource limitations, Titanic implements fine-tuning

of LLMs on client devices in four ways: (i) Model partitioning and distributed fine-tuning: Titanic splits the LLM across multiple client devices for fine-tuning instead of requiring each client to train the entire model independently. This significantly reduces the computational burden on individual clients, allowing resource-constrained devices to participate in model training. (ii) Optimized client selection: Titanic first selects a subset of clients using an efficient integer optimization algorithm. These clients are more representative in terms of computational resources and data quality. In this way, Titanic ensures that the participating clients can efficiently complete the tasks assigned to them while reducing over-reliance on individual client resources. (iii) Reduced communication overhead: Titanic significantly reduces communication costs by transmitting only a small number of model weights between clients instead of entire model updates. This not only protects privacy but also lowers bandwidth requirements, making distributed training more feasible. (iv) Model-agnostic partitioning mechanism: Focusing on feasibility, Titanic adopts a model-agnostic partitioning mechanism that can fully automate the splitting and distribution of any LLM to client devices. This means that Titanic can flexibly adapt to different models and trainers without modifying the model source code. Through these technical means, Titanic effectively addresses the problem of client resource limitations while protecting data privacy, making it possible to fine-tune LLMs on resource-constrained devices.

The privacy protection solutions for FLLM are still in their infancy, and the specific problems they target vary. To enhance the security of FLLM, there is an urgent need for deeper and broader exploration in this area.

4.2 Security threats and defenses

Table 5. Overview of security in FLLM.

Type		Describe	Approaches
Attacks	Data Poisoning Attack	Maliciously alter training data to degrade model performance.	-
	Model Poisoning Attack	Maliciously modify model parameters or updates to degrade the overall model performance.	-
	Backdoor Attack	Insert hidden triggers into the LLM to produce incorrect outputs when activated.	[53], [67], [115], [54]
Defenses	Distance based defense	Adversaries are identified by the distance deviations in malicious updates from normal ones.	[153], [92], [147]
	Feature based defense	Maliciously tampered model updates exhibit distinct characteristics from benign updates in certain features.	[69], [8], [144], [3]
	Knowledge Distillation	Integrate clustering, model selection, and knowledge distillation to identify and filter malicious client updates.	[2]

4.2.1 Data Poisoning Attack. Data Poisoning Attacks (DPA) occur during the data collection phase on the client side, where the original data is modified to train a poisoned local model, which is then uploaded to participate in aggregation to harm the global model and compromise its availability or integrity. Shafahi et al. [93] explored an optimization-based "clean-label" data poisoning attack on neural networks, which manipulates the model's behavior at test time by adding carefully designed samples to the training set. This type of attack does not require the attacker to control the labels of

the training data but instead leverages the model's "memory" of the data during training to achieve its goals. Another form of DPA is the label-flipping attack, where the adversary modifies the labels of the dataset rather than the sample features to generate a poisoned model. Tolpegin et al. [107] studied label-flipping attacks on FL systems, demonstrating that even a small number of malicious participants can significantly reduce classification accuracy and recall, and that the attack can be targeted to negatively impact specific categories. LLMs, with their higher complexity and stronger fitting capabilities, can capture subtle features in the data, including maliciously altered label information. Therefore, label-flipping attacks may have a more pronounced impact on LLMs, as they are more prone to overfitting incorrect label information. Shejwalkar et al. [95] systematically analyzed various possible threat models, variants of poisoning attacks, and different capabilities of attackers, with a particular focus on non-targeted poisoning attacks. They found that, contrary to common belief, FL shows high robustness in practical applications even with simple and low-cost defense measures. Based on this, they proposed new state-of-the-art data poisoning attack methods and demonstrated their ineffectiveness in the presence of simple defense mechanisms through extensive experiments on three benchmark datasets. In addition to directly modifying the original data, Zhang et al. [140] proposed PoisonGAN, a generative poisoning attack model for FL systems. This method, based on generative adversarial networks (GANs), uses the parameters of the global model to generate toxic data samples that mimic the training samples of other participants and forge the labels of these samples. Since the federated fine-tuning process typically involves making a small number of updates to the pre-trained model, these toxic data samples can significantly impact the model's performance during the fine-tuning stage.

4.2.2 Model Poisoning Attack. Unlike DPA, Model Poisoning Attacks (MPA) occur during the training phase on the client side, where the local model is modified to achieve the goal of corrupting the global training. Fang et al. [24] first systematically studied MPA and formalized the attack problem as an optimization problem, targeting four byzantine fault-tolerant FL defense methods. This optimization approach helps minimize the difference between the current poisoned model and the model from the previous round, making it more difficult for the server to detect the attack. Bagdasaryan et al. [4] used a model replacement method to blend the poisoned model with a benign model and employed hyperparameter scaling to evade detection. Shejwalkar et al. [94] proposed a more effective model poisoning attack, similar to the Min-Max attack, which constrains the upper bound of the sum of squared distances between the malicious gradient and all benign gradients to be the sum of squared distances between any benign gradient and other benign gradients, thereby ensuring the survival rate of the malicious model. Federated fine-tuning typically involves making a small number of updates to the parameters of the pre-trained model, based on the local data of the clients. Attackers can tamper with these updates to directly affect the fine-tuning process of the global model. However, MPA targeting FLLM have not yet emerged.

4.2.3 Backdoor Attack. Poisoning attacks degrade the performance of the global model by tampering with data and models, while backdoor attacks (BA) manipulate model behavior by injecting specific attack information or data. Li et al. [53] investigated the threat of backdoor attacks when fine-tuning base models in FL, proposing a method to embed backdoors into the base model and transfer them into the FL system. This allows the successful implantation of backdoors in the global model without fully participating in the FL process. Yang et al. [125] explored vulnerabilities to backdoor attacks in the word embedding layer of natural language processing models. They found that attackers could inject backdoors by modifying only one word embedding vector (i.e., the embedding vector of the trigger word) without accessing the target dataset. This enables the model to produce incorrect classifications for input samples containing the specific trigger word without affecting its performance on normal samples. However, the trigger word

needs to be rare and not appear in the clean test set, which may limit the practical application scenarios of the attack, as attackers need to carefully select trigger words to avoid detection. Yoo et al. [133] studied the feasibility of backdoor attacks through rare word embeddings and gradient ensembling. Attackers can inject backdoors by manipulating the embedding vectors of rare words, causing the model to produce incorrect outputs for inputs containing specific trigger words without affecting its performance on normal samples. Lyu et al. [67] proposed a novel backdoor attack method called PFedBA, which optimizes the trigger generation process to align the gradients and losses of the backdoor task with the main task, embedding undetectable backdoors in personalized models. Wu et al. [115] introduced a novel attack strategy that generates synthetic data on the server side using a tampered base model and implants backdoors during client model initialization and knowledge distillation. This attack method has a high success rate in various image and text classification tasks, and existing FL defense strategies have limited effectiveness against this novel attack. Xi et al. [54] proposed a new backdoor attack called Fed-EBD, which generates backdoored synthetic data on the server side using a tampered base model and propagates it to client models without requiring the attacker to fully control clients or continuously participate in the FL process. Experiments showed that this attack has a high success rate in various heterogeneous FL configurations and benchmark datasets and can effectively evade existing backdoor defense strategies. The study revealed significant security risks when using federated models in horizontal FL and emphasized the urgency of developing more robust defense mechanisms.

However, current backdoor attacks generally have strong attack assumptions and limited experimental scopes, with insufficient consideration of practical application scenarios. They do not fully account for the potential impact of other security mechanisms (such as client authentication and data encryption) that may exist in real systems on the attacks.

4.2.4 Defenses. To achieve robust evaluation of model parameters from client models, researchers have conducted more in-depth explorations. Zhou et al. [153] proposed SecFFT, which utilizes frequency-domain transformations to extract the low-frequency components of model updates and identifies malicious updates inconsistent with the normal update distribution using chi-square distance. It also analyzes the historical behavior sequences of nodes to construct attack intentions and employs the local outlier factor algorithm to identify malicious intentions hidden behind seemingly normal behaviors. By combining these two methods, SecFFT can effectively detect complex and covert backdoor attacks while maintaining high performance and robustness in federated fine-tuning. Ma et al. [69] proposed a classifier based on persistent homology and persistent graphs, which identifies malicious clients by analyzing the topological features of neural network models. This method can efficiently detect various types of backdoor attacks even under highly imbalanced non-i.i.d. data conditions. Basak et al. [8] proposed the DPAD scheme, which uses an auditing mechanism to check the integrity and consistency of client updates, identifying potentially maliciously tampered data or abnormal behaviors to prevent these harmful updates from affecting the global model. Ren et al. [92] proposed the BPFL method, which detects malicious behavior by calculating the cosine similarity between client local gradients and global gradients. Malicious gradients typically have lower similarity to the global gradient, so this similarity calculation can identify tampered gradients and prevent them from causing damage to the global model.

To enhance the robustness of FLLM systems against poisoning and backdoor attacks, researchers have successively proposed targeted defense schemes. Zhang et al. [144] proposed the Fed-FA backdoor attack defense algorithm, which uses the f-divergence metric to estimate the differences in client data and addresses the issue of client data invisibility through a Hessian redistribution mechanism in the synthetic dataset and embedding layer. It demonstrates how to detect and exclude suspicious clients by modeling the differences in client data distributions, thereby effectively defending against backdoor attacks. However, the complexity of calculating the Hessian matrix and the f-divergence metric is

high, and the defense capability for non-i.i.d. data is limited. Ali et al. [3] proposed a new defense mechanism called AGSD, which detects malicious client model updates by identifying adversarial biases and overly confident predictions in the attacked model. It combines clustering algorithms and client trust history to select the most trustworthy client updates for model aggregation. AGSD can effectively defend against attacks even with a very small retained dataset ($\leq$ 0.1% training data) or when using out-of-distribution data, with minimal impact on the accuracy of clean data. However, the defense effectiveness of AGSD against certain specific types of adaptive attacks (such as low-confidence backdoor attacks) still needs further verification. Alharbi et al. [2] proposed the RKD (Robust Knowledge Distillation) defense mechanism, which identifies and filters out malicious client updates by combining clustering, model selection, and knowledge distillation techniques. This approach constructs a reliable model ensemble and distills the knowledge of these models into the global model. However, it requires additional computational resources, especially in large-scale FL environments. Zhang et al. [147] proposed Dim-Krum, noting that backdoor attacks in NLP are more difficult to defend against compared to the computer vision (CV) domain. This is because NLP attacks typically have lower relative backdoor strengths, leading to poor performance of existing robust federated aggregation methods in NLP tasks. To address this, the authors proposed an improved algorithm Dim-Krum based on the Krum framework, which calculates distances between clients only in a few dimensions. This effectively detects and discards malicious updates, significantly reducing the success rate of backdoor attacks while maintaining high accuracy on clean data.

5 Future Directions

As FLLM continue to improve in terms of feasibility, robustness, and security, user demands are also increasing. For example, joint training under few-shot conditions, the need for unlearning techniques when users withdraw their private data or exit the system, and the protection of model IP rights by the server. These issues pose challenges to the sustainable development of FLLM.

5.1 Few-shot learning in FLLM

Existing FL methods, when dealing with LLMs, often require a substantial amount of labeled data to achieve effective fine-tuning, which is neither economical nor feasible in practical applications. Particularly in fields such as healthcare and finance, where data privacy and security are of utmost importance, data is often difficult to label and share on a large scale. Few-shot learning (FSL) techniques have made significant progress in reducing the demand for labeled data, but integrating them with FL to adapt to LLMs still faces many challenges. On the one hand, FSL relies on the model's ability to learn from context, and the distributed nature of FL may lead to insufficient knowledge transfer between clients, thereby affecting the effectiveness of FSL. On the other hand, how to efficiently generate and utilize a small amount of labeled data within the federated framework to enhance model performance remains an urgent issue to be resolved. Therefore, how to fully leverage the potential of decentralized data while protecting data privacy and reducing dependence on large-scale labeled data has become a key issue in current research on FLLM.

Existing instruction fine-tuning methods typically assume that clients already possess structured instruction-response pair data, which is unrealistic in practice because client data is usually unstructured text. Therefore, manually annotating this data is not only time-consuming but also limits the widespread application of federated instruction fine-tuning. Ye et al. [130] proposed the FedIT-U2S framework, which leverages few-shot prompting techniques to combine unstructured text and a small number of examples to automatically generate structured instruction-response pair data, as shown in Figure 7. It also introduces a retrieval-based example selection technique that automatically selects examples based on the relevance between client data and the example pool, avoiding the complexity of manually selecting examples.

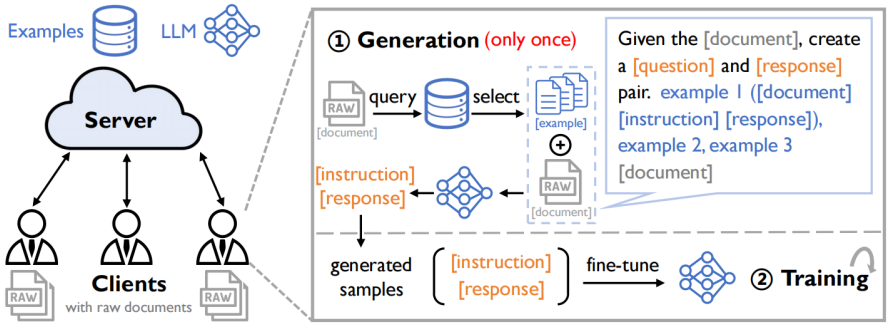

Fig. 7. FedIT-U2S workflow from [130]. FedIT-U2S employs a retrieval-based example selection technique to automatically select the most relevant examples for each client's unstructured data from an example pool provided by the server. These examples are then combined with the client's unstructured data pieces to form prompts, which are fed into a pre-trained language model to generate instruction-response pairs. This process transforms the unstructured data into structured instruction-tuning data.

Although FedIT-U2S reduces the need for manually annotated data and expands the application scope of federated instruction fine-tuning, the quality of the generated data lacks an excellent evaluation mechanism for screening, and its performance is highly dependent on the base model. Cai et al. [17] proposed the AUG-FedPrompt scheme, which designs a comprehensive data generator to simulate the distribution of labeled data in few-shot learning tasks and combines prompt learning and pseudo-labeling techniques to augment data using a large amount of unlabeled data. However, while this method achieves high performance, it also brings significant system overhead, including high computational latency, large memory requirements, and high communication costs, and its robustness to LLMs is debatable. Building on this, to address the issue of scarce and unevenly distributed data labels, they proposed the FeS framework [14]. Unlike [17], FeS focuses more on system-level optimization, significantly reducing training latency, device energy consumption, and network traffic through curriculum training progress control, representative diversity filtering, and co-planning of training depth and layer capacity, while maintaining model performance comparable to full dataset fine-tuning [14]. Jiang et al. [39] proposed the LP-FL framework, which guides LLMs to understand task objectives by adding task descriptions to input samples. This allows clients to leverage the global model's knowledge to assign soft labels to unlabeled data and gradually incorporate it into the training set, thereby dynamically expanding the labeled dataset during the FL process.

5.2 FLLM Unlearning

The goal of machine learning is to extract knowledge from data, while machine unlearning endows models with the ability to "forget" specific data. Its core lies in adjusting model parameters to achieve the effect of certain data not participating in training, thereby avoiding retraining [70]. This approach allows the server to remove the contributions of specific user data according to user requests, ensuring that the model cannot trace these data, thereby protecting privacy. Meanwhile, model updates resulting from forgetting erroneous or low-quality data can further enhance model security.

In FL, simply removing the updates of the target user from the global model is not sufficient, as other users' historical models still retain the data to be forgotten, and these data will be re-aggregated in subsequent training. Large-scale data forgetting may lead to catastrophic forgetting [62] significantly reducing model performance. Zhang et al. [141] proposed CGKD, which is specifically designed to address the model recovery issue in federated unlearning, particularly

under scenarios with limited server data resources. CGKD constructs the unlearning model by erasing all historical contributions of the target client and treats it as the student model. It then fine-tunes the pre-trained CLIP model using a small number of samples on the server side to generate a more robust teacher model. During the fine-tuning process, the original backbone network of CLIP is kept intact, and an adapter module is introduced to dynamically integrate the fine-tuned features with the original features through residual connections, thereby enhancing the model's understanding of the semantic context of images. This approach effectively mitigates the negative impact of unlearning operations on model performance. Zuo et al. [160] proposed a blockchain-based federated learning framework for LLMs, which leverages the tamper-proof and distributed ledger features of blockchain to create an immutable record of each model's contributions, thereby enhancing transparency and accountability. This function is seamlessly integrated with the federated learning mechanism, allowing data owners to remove their data from the training process while minimizing the impact on other participants. The mechanism is implemented through blockchain's smart contracts, ensuring the security and transparency of the unlearning process. Liu et al. [64] proposed an efficient federated unlearning method called Rapid Retraining, which uses fast retraining and a distributed Newton-type update algorithm. It leverages the diagonal empirical Fisher information matrix to approximate the inverse Hessian vector and introduces momentum techniques to achieve data deletion while reducing errors and enhancing model utility. This method is model agnostic and can be combined with the optimization techniques commonly used in federated fine-tuning (such as LoRA), and it is possible to implement it in LLMs. Su et al. [100] proposed a novel asynchronous federated unlearning mechanism called KNOT, which divides clients into multiple clusters and performs aggregation only within each cluster. This approach confines the retraining caused by data deletion to within the cluster. To optimize the assignment of clients to clusters, the authors formulated the problem as a solvable optimization problem, namely the lexicographic minimization problem, and demonstrated that it can be efficiently solved using a linear programming solver, significantly reducing time overhead. This asynchronous federated learning method can improve efficiency in FLLM. However, although clustering reduces the number of clients that need to be retrained, the cost may still be high in the FLLM scenario. Zhu et al. [158] proposed the FedLU framework for learning and offloading heterogeneous knowledge graph embeddings. Based on cognitive neuroscience theory, they proposed an offloading method that combines retroactive interference and passive decay. This method can delete specific knowledge from both local clients and the global model without significantly affecting overall performance, meeting the needs of privacy protection and data deletion. This method is promising for FLLM. Clients can utilize their local data to fine-tune pre-trained LLMs and then transfer the local knowledge to the global model through knowledge distillation.

For complex LLMs, fine-tuning is prone to catastrophic forgetting, which is the loss of old knowledge when learning new tasks [138]. This limits the generality and scalability of multi-task learning. Zhu et al. [154] introduced a post-training adjustment method called "Model Tailor." This method retains the pre-trained parameters of LLMs while replacing a small portion ($\leq 10\%$) of the fine-tuning parameters. Model Tailor employs a second-order analysis-based approach to evaluate the importance of each parameter and selectively modifies those parameters that have the least impact on both the target and original tasks, thereby ensuring that the model retains most of the pre-trained knowledge after fine-tuning. Li et al. [44] revealed a direct link between the flatness of the model loss landscape and the degree of catastrophic forgetting. Based on this connection, they introduced the Sharpness-Aware Minimization method to flatten the optimization landscape, attempting to maintain the model's memory of previous knowledge during fine-tuning, thereby alleviating the model forgetting issue. Lee et al. [41] proposed a new method called Base-Anchored Preference Optimization. The core idea of BAPO is to maintain the possibility of the policy model generating base responses originating from the reference model during the process of personalized preference optimization. By introducing an

anchoring mechanism for base responses in the optimization process, BAPO can ensure that the policy model does not lose the knowledge contained in the base responses when adapting to different user preferences, thereby effectively alleviating the problem of knowledge forgetting. These unlearning methods for LLMs may be inefficient in FLLM, as when a client initiates a request to forget data, the server cannot simply unlearn on the global model alone but also needs to ensure that the historical models of other clients forget this data as well. This can likely only be achieved through multiple iterations slowly, posing a significant challenge to the efficiency of unlearning.

Although there has been much research on machine unlearning in FL and LLMs, there is still a gap in the emerging issue of data forgetting in FLLM. Current federated unlearning methods are not only computationally and communicatively expensive for LLMs but also difficult to transplant to LoRA and adapter architectures. How to implement federated unlearning schemes for LLMs while avoiding catastrophic forgetting is of high research value in the coming period.

5.3 IP Protection in FLLM

Given the high training costs of LLMs, managing the authorized use of models becomes particularly crucial, and cost-effective model watermarking techniques can protect model intellectual property (IP) rights and prevent models from being illegally copied or misused. Liu et al. [59] summarized text watermarking techniques for LLMs and found that the advanced semantic understanding and context-aware capabilities of LLMs make watermark embedding more covert while reducing the impact on the original text semantics. Embedding watermarks in LLM-generated text can effectively track and detect LLM-generated text, helping to control potential misuse.

Yang et al. [43] proposed the FedIPR framework, which allows users to independently embed private watermarks in their local models and verify these watermarks after model aggregation to prove IP rights over the federated model. FedIPR implements feature-based watermarking, embedding binary strings in the parameters of the model's normalization layers as watermarks, and backdoor-based watermarking, introducing specific trigger samples (such as adversarial samples) during model training so that the model outputs specific incorrect labels when receiving these trigger samples, thereby verifying ownership.

LLMs typically have complex structures and a large number of parameters, providing more space for watermark embedding. For example, the normalization layers of the transformer architecture can be used for feature-based watermark embedding, while adversarial samples can serve as triggers for backdoor-based watermarking. Liao et al. [56] noted that in heterogeneous FL, watermarks embedded in the global model may be damaged to varying degrees when transferred to users' heterogeneous models, failing to provide complete ownership protection in local models. Therefore, they proposed the PWFed method to protect model IP rights in heterogeneous FL. PWFed uses GAN technology to generate dynamic watermark samples that are indistinguishable from original samples and designs two different granularity watermark embedding strategies to ensure the robustness and stealth of watermarks in personalized models. However, in the context of LLMs, PWFed may require greater computational overhead, and its robustness remains to be considered.

So far, there has been a considerable amount of literature on IP protection for FL and LLMs, but there is still much room for exploration regarding the emerging FLLM. In the FL environment, how to ensure the legal and authorized use of LLMs and prevent unauthorized copying and dissemination is an urgent issue to be addressed.

6 Conclusion

With the increasing popularity of LLMs among the general public, the demand for training data has surged exponentially, while public data resources are gradually being depleted. Directly using users' private data for training would severely violate privacy. Against this backdrop, the integration of LLMs and FL has emerged and is gaining increasing attention, with the potential for broader applications in the future. However, the field of FLLM is still in its infancy, with key issues that need to be addressed urgently. Starting from the temporal overhead, heterogeneity, security and privacy issues, and other special issues of FLLM, we have discussed the cutting-edge research on the feasibility, robustness, security, and future directions of current FLLM and found the following characteristics:

1. Research on the feasibility of FLLM has become increasingly sophisticated academically but still has a significant gap from practical application. The computational overhead of client training and the communication overhead required for transmitting models during federated fine-tuning of LLMs can be reduced by hundreds or even thousands of times compared to full-parameter tuning through PEFT and some special methods. However, due to the large base of these overheads, even with such reductions, it remains challenging for ordinary participants to train LLMs on local devices. In the future, with the continuous optimization of computing resources and the development of distributed training technologies, the fine-tuning efficiency of FLLM is expected to improve further, especially with efficient training and optimization in large-scale distributed environments becoming a research focus.

2. Research on the robustness of FLLM is being actively conducted but still needs to address challenges from multiple aspects. The current research mainly focuses on resource heterogeneity, due to the high resource threshold for training LLMs and the significant differences in training capabilities among users, which easily leads to resource heterogeneity. For data and task heterogeneity issues, there are currently few solutions, and it is evident that most of them are based on LoRA and Adapter. In the future, the robustness of FLLM will gradually become a research focus after feasibility, and solving various realistic heterogeneity issues will be a key step for the practical application of FLLM.

3. Research on the security of FLLM has just begun and urgently needs to focus on threats and defenses related to privacy and security. Current research on the security of FLLM mainly focuses on backdoor attacks, while other potential threats such as data reconstruction attacks, jailbreaking attacks, and poisoning attacks are still very rare. We have found that due to different fine-tuning methods in FLLM training, the related attack and defense schemes also vary. Currently, there is limited research on various attack and defense schemes targeting different fine-tuning methods of FLLM. It can be anticipated that the field of privacy and security for FLLM will exhibit a wide range of involvement and diverse solutions.

4. Future directions of FLLM holds significant potential. For instance, the few-shot learning problem is particularly salient for FLLM, which necessitates a substantial amount of training data; federated unlearning of LLMs serves as a safeguard for users' rights to delete data; and the issue of IP protection for FLLM profoundly affects the enthusiasm of all parties involved in the training process. Research on these aspects of FLLM is currently nascent, with only a handful of papers addressing these issues. Moving forward, as these technologies continue to evolve and converge, FLLM is anticipated to better accommodate diverse complex scenarios, thereby facilitating sustainable development.

By further exploring the synergistic relationship between FL and LLMs, the field of FLLM can be advanced, leading to the development of more efficient, effective, secure, privacy-preserving, and personalized LLMs. This integration has the potential to transform artificial intelligence across various fields and promote the deployment of powerful and ethically responsible advanced AI systems.

References

[1] Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. 2022. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *Advances in neural information processing systems* 35 (2022), 29677–29690.

[2] Ebtisaam Alharbi, Leandro Soriano Marcolino, Qiang Ni, and Antonios Gouglidis. 2025. Robust Knowledge Distillation in Federated Learning: Counteracting Backdoor Attacks. *CoRR* abs/2502.00587 (2025). doi:10.48550/ARXIV.2502.00587 arXiv:2502.00587

[3] Hassan Ali, Surya Nepal, Salil S. Kanhere, and Sanjay K. Jha. 2024. Adversarially Guided Stateful Defense Against Backdoor Attacks in Federated Deep Learning. *CoRR* abs/2410.11205 (2024). doi:10.48550/ARXIV.2410.11205 arXiv:2410.11205

[4] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*. PMLR, 2938–2948.

[5] Guangji Bai, Yijiang Li, Zilinghan Li, Liang Zhao, and Kibaek Kim. 2024. FedSpaLLM: Federated Pruning of Large Language Models. *CoRR* abs/2410.14852 (2024). doi:10.48550/ARXIV.2410.14852 arXiv:2410.14852

[6] Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. 2024. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[7] Li Bai, Haibo Hu, Qingqing Ye, Haoyang Li, Leixia Wang, and Jianliang Xu. 2024. Membership Inference Attacks and Defenses in Federated Learning: A Survey. *Comput. Surveys* 57, 4 (2024), 1–35.

[8] Santanu Basak and Kakali Chatterjee. 2025. DPAD: Data Poisoning Attack Defense Mechanism for federated learning-based system. *Computers and Electrical Engineering* 121 (2025), 109893.

[9] Hajira Batool, Adeel Anjum, Abid Khan, Stefano Izzo, Carlo Mazzocca, and Gwanggil Jeon. 2024. A secure and privacy preserved infrastructure for VANETs based on federated learning with local differential privacy. *Information Sciences* 652 (2024), 119717.

[10] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *International conference on machine learning*. PMLR, 634–643.

[11] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2013), 387–402. Issue 3.

[12] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *CoRR* abs/2108.07258 (2021). arXiv:2108.07258 https://arxiv.org/abs/2108.07258

[13] Dongqi Cai, Shangguang Wang, Yaozong Wu, Felix Xiaozhu Lin, and Mengwei Xu. 2023. Federated Few-Shot Learning for Mobile NLP. In *29th Annual International Conference on Mobile Computing and Networking, MobiCom 2023*.

[14] Dongqi Cai, Shangguang Wang, Yaozong Wu, Felix Xiaozhu Lin, and Mengwei Xu. 2023. Federated few-shot learning for mobile nlp. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–17.

[15] Dongqi Cai, Yaozong Wu, Shangguang Wang, Felix Xiaozhu Lin, and Mengwei Xu. 2022. Fedadapter: Efficient federated learning for modern nlp. *arXiv preprint arXiv:2205.10162* (2022).

[16] Dongqi Cai, Yaozong Wu, Shangguang Wang, Felix Xiaozhu Lin, and Mengwei Xu. 2023. Efficient federated learning for modern nlp. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–16.

[17] Dongqi Cai, Yaozong Wu, Haitao Yuan, Shangguang Wang, Felix Xiaozhu Lin, and Mengwei Xu. 2023. Towards practical few-shot federated nlp. In *Proceedings of the 3rd Workshop on Machine Learning and Systems*. 42–48.

[18] Chaochao Chen, Xiaohua Feng, Jun Zhou, Jianwei Yin, and Xiaolin Zheng. 2023. Federated Large Language Model: A Position Paper. *CoRR* abs/2307.08925 (2023). doi:10.48550/ARXIV.2307.08925 arXiv:2307.08925

[19] Haokun Chen, Yao Zhang, Denis Krompass, Jindong Gu, and Volker Tresp. 2024. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 11285–11293.

[20] Yi-Qiang Chen, Teng Zhang, Xin-Long Jiang, Qian Chen, Chen-Long Gao, and Wu-Liang Huang. 2024. Fedbone: Towards large-scale federated multi-task learning. *Journal of Computer Science and Technology* 39, 5 (2024), 1040–1057.

[21] Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. 2024. Heterogeneous lora for federated fine-tuning of on-device foundation models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 12903–12913.

[22] Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do Membership Inference Attacks Work on Large Language Models? *CoRR* abs/2402.07841 (2024). doi:10.48550/ARXIV.2402.07841 arXiv:2402.07841

[23] Alexander V Eriksen, Sören Möller, and Jesper Ryg. 2024. Use of GPT-4 to diagnose complex clinical cases. AIp2300031 pages.

[24] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. 2020. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)*. 1605–1622.

[25] Zihan Fang, Zheng Lin, Zhe Chen, Xianhao Chen, Yue Gao, and Yuguang Fang. 2024. Automated Federated Pipeline for Parameter-Efficient Fine-Tuning of Large Language Models. *CoRR* abs/2404.06448 (2024). doi:10.48550/ARXIV.2404.06448 arXiv:2404.06448

[26] Liam H. Fowl, Jonas Geiping, Wojciech Czaja, Micah Goldblum, and Tom Goldstein. 2022. Robbing the Fed: Directly Obtaining Private Data in Federated Learning with Modified Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29,*

2022. OpenReview.net. https://openreview.net/forum?id=fwzUgo0FM9v

[27] Liam H. Fowl, Jonas Geiping, Steven Reich, Yuxin Wen, Wojciech Czaja, Micah Goldblum, and Tom Goldstein. 2023. Decepticons: Corrupted Transformers Breach Privacy in Federated Learning for Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=r0BrY4BiEXO

[28] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. 2020. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*. 301–316.

[29] Tao Guo, Song Guo, Junxiao Wang, Xueyang Tang, and Wenchao Xu. 2023. Promptfl: Let federated participants cooperatively learn prompts instead of models-federated learning in age of foundation model. *IEEE Transactions on Mobile Computing* (2023).

[30] Umang Gupta, Dimitris Stripelis, Pradeep K Lam, Paul Thompson, Jose Luis Ambite, and Greg Ver Steeg. 2021. Membership inference attacks on deep regression models for neuroimaging. In *Medical Imaging with Deep Learning*. PMLR, 228–251.

[31] Mengde Han, Tianqing Zhu, and Wanlei Zhou. 2024. Fair Federated Learning with Opposite GAN. *Knowledge-Based Systems* (2024), 111420. Issue No.C.

[32] Shanshan Han, Baturalp Buyukates, Zijian Hu, Han Jin, Weizhao Jin, Lichao Sun, Xiaoyang Wang, Wenxuan Wu, Chulin Xie, Yuhang Yao, et al. 2024. Fedsecurity: A benchmark for attacks and defenses in federated learning and federated llms. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5070–5081.

[33] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*. PMLR, 2790–2799.

[34] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=nZeVKeeFYf9

[35] Jiahui Hu, Dan Wang, Zhibo Wang, Xiaoyi Pang, Huiyu Xu, Ju Ren, and Kui Ren. 2024. Federated Large Language Model: Solutions, Challenges and Future Directions. *IEEE Wireless Communications* (2024).

[36] Teodor Ivănușcă and Cosmin-Iulian Irimia. 2024. The Impact of Prompting Techniques on the Security of the LLMs and the Systems to Which They Belong. *Applied Sciences (2076-3417)* 14, 19 (2024).

[37] Ruofan Jia, Weiying Xie, Jie Lei, Haonan Qin, Jitao Ma, and Leyuan Fang. 2024. Towards Efficient Model-Heterogeneity Federated Learning for Large Models. *CoRR* abs/2411.16796 (2024). doi:10.48550/ARXIV.2411.16796 arXiv:2411.16796

[38] Feibo Jiang, Li Dong, Siwei Tu, Yubo Peng, Kezhi Wang, Kun Yang, Cunhua Pan, and Dusit Niyato. 2024. Personalized Wireless Federated Learning for Large Language Models. *CoRR* abs/2404.13238 (2024). doi:10.48550/ARXIV.2404.13238 arXiv:2404.13238

[39] Jingang Jiang, Haiqi Jiang, Yuhan Ma, Xiangyang Liu, and Chenyou Fan. 2024. Low-parameter federated learning with large language models. In *International Conference on Web Information Systems and Applications*. Springer, 319–330.

[40] Shuyu Jiang, Xingshu Chen, Kaiyu Xu, Liangguo Chen, Hao Ren, and Rui Tang. 2025. Decomposition, Synthesis and Attack: A Multi-Instruction Fusion Method for Jailbreaking LLMs. *IEEE Internet of Things Journal* (2025).

[41] Gihun Lee, Minchan Jeong, Yujin Kim, Hojung Jung, Jaehoon Oh, SangMook Kim, and Se-Young Yun. 2024. BAPO: Base-Anchored Preference Optimization for Overcoming Forgetting in Large Language Models Personalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, 6804–6820. https://aclanthology.org/2024.findings-emnlp.398

[42] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 3045–3059. doi:10.18653/V1/2021.EMNLP-MAIN.243

[43] Bowen Li, Lixin Fan, Hanlin Gu, Jie Li, and Qiang Yang. 2022. FedIPR: Ownership verification for federated deep neural network models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 4 (2022), 4521–4536.

[44] Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024. Revisiting Catastrophic Forgetting in Large Language Model Tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, 4297–4308. https://aclanthology.org/2024.findings-emnlp.249

[45] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. 2023. Effective passive membership inference attacks in federated learning against overparameterized models. In *The Eleventh International Conference on Learning Representations*.

[46] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems* 34 (2021), 9694–9705.

[47] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 965–978.

[48] Shenghui Li, Edith C-H Ngai, and Thiemo Voigt. 2023. An experimental study of byzantine-robust aggregation schemes in federated learning. *IEEE Transactions on Big Data* (2023).

[49] Shenghui Li, Edith C. H. Ngai, Fanghua Ye, and Thiemo Voigt. 2024. PEFT-as-an-Attack! Jailbreaking Language Models during Federated Parameter-Efficient Fine-Tuning. *CoRR* abs/2411.19335 (2024). doi:10.48550/ARXIV.2411.19335 arXiv:2411.19335

[50] Shenghui Li, Fanghua Ye, Meng Fang, Jiaxu Zhao, Yun-Hin Chan, Edith C. H. Ngai, and Thiemo Voigt. 2024. Synergizing Foundation Models and Federated Learning: A Survey. *CoRR* abs/2406.12844 (2024). doi:10.48550/ARXIV.2406.12844 arXiv:2406.12844

[51] Xingyu Li, Lu Peng, Yu-Ping Wang, and Weihua Zhang. 2025. Open challenges and opportunities in federated foundation models towards biomedical healthcare. *BioData Min.* 18, 1 (2025). doi:10.1186/S13040-024-00414-9

[52] Xi Li and Jiaqi Wang. 2024. Position Paper: Assessing Robustness, Privacy, and Fairness in Federated Learning Integrated with Foundation Models. *CoRR* abs/2402.01857 (2024). doi:10.48550/ARXIV.2402.01857 arXiv:2402.01857

[53] Xi Li, Songhe Wang, Chen Wu, Hao Zhou, and Jiaqi Wang. 2023. Backdoor Threats from Compromised Foundation Models to Federated Learning. *CoRR* abs/2311.00144 (2023). doi:10.48550/ARXIV.2311.00144 arXiv:2311.00144

[54] Xi Li, Chen Wu, and Jiaqi Wang. 2024. Unveiling backdoor risks brought by foundation models in heterogeneous federated learning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 168–181.

[55] Zhiwei Li and Guodong Long. 2024. Navigating the Future of Federated Recommendation Systems with Foundation Models. *CoRR* abs/2406.00004 (2024). doi:10.48550/ARXIV.2406.00004 arXiv:2406.00004

[56] Yuying Liao, Rong Jiang, and Bin Zhou. 2024. Dynamic Black-Box Model Watermarking for Heterogeneous Federated Learning. *Electronics* 13, 21 (2024), 4306.

[57] Zheng Lin, Xuanjie Hu, Yuxin Zhang, Zhe Chen, Zihan Fang, Xianhao Chen, Ang Li, Praneeth Vepakomma, and Yue Gao. 2024. SplitLoRA: A Split Parameter-Efficient Fine-Tuning Framework for Large Language Models. *CoRR* abs/2407.00952 (2024). doi:10.48550/ARXIV.2407.00952 arXiv:2407.00952

[58] Zhenqing Ling, Daoyuan Chen, Liuyi Yao, Yaliang Li, and Ying Shen. 2024. On the convergence of zeroth-order federated tuning for large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1827–1838.

[59] Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip Yu. 2024. A survey of text watermarking in the era of large language models. *Comput. Surveys* 57, 2 (2024), 1–36.

[60] Bowen Liu, Boao Xiao, Xutong Jiang, Siyuan Cen, Xin He, and Wanchun Dou. 2023. Adversarial Attacks on Large Language Model-Based System and Mitigating Strategies: A Case Study on ChatGPT. *Security & Communication Networks* (2023).

[61] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks. *CoRR* abs/2110.07602 (2021). arXiv:2110.07602 https://arxiv.org/abs/2110.07602

[62] Yang Liu, Mingyuan Fan, Cen Chen, Ximeng Liu, Zhuo Ma, Li Wang, and Jianfeng Ma. 2022. Backdoor defense with machine unlearning. In *IEEE INFOCOM 2022-IEEE conference on computer communications*. IEEE, 280–289.

[63] Yuxi Liu, Guibo Luo, and Yuesheng Zhu. 2024. FedFMS: Exploring Federated Foundation Models for Medical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 283–293.

[64] Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. 2022. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022-IEEE conference on computer communications*. IEEE, 1749–1758.

[65] Jiahao Lu, Xi Sheryl Zhang, Tianli Zhao, Xiangyu He, and Jian Cheng. 2022. April: Finding the achilles' heel on privacy for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10051–10060.

[66] Wang Lu, Hao Yu, Jindong Wang, Damien Teney, Haohan Wang, Yiqiang Chen, Qiang Yang, Xing Xie, and Xiangyang Ji. 2023. ZooPFL: Exploring Black-box Foundation Models for Personalized Federated Learning. *CoRR* abs/2310.05143 (2023). doi:10.48550/ARXIV.2310.05143 arXiv:2310.05143

[67] Xiaoting Lyu, Yufei Han, Wei Wang, Jingkai Liu, Yongsheng Zhu, Guangquan Xu, Jiqiang Liu, and Xiangliang Zhang. 2024. Lurking in the shadows: Unveiling stealthy backdoor attacks against personalized federated learning. In *33rd USENIX Security Symposium (USENIX Security 24)*. 4157–4174.

[68] Xiaodong Ma, Jia Zhu, Zhihao Lin, Shanxuan Chen, and Yangjie Qin. 2022. A state-of-the-art survey on solving non-iid data in federated learning. *Future Generation Computer Systems* 135 (2022), 244–258.

[69] Zihan Ma and Tianchong Gao. 2024. Federated learning backdoor attack detection with persistence diagram. *Computers & Security* 136 (2024), 103557.

[70] Zhuo Ma, Yang Liu, Ximeng Liu, Jian Liu, Jianfeng Ma, and Kui Ren. 2022. Learn to forget: Machine unlearning via neuron masking. *IEEE Transactions on Dependable and Secure Computing* 20, 4 (2022), 3194–3207.

[71] Shubham Malaviya, Manish Shukla, and Sachin Lodha. 2023. Reducing communication overhead in federated learning for pre-trained language models using parameter-efficient finetuning. In *Conference on Lifelong Learning Agents*. PMLR, 456–469.

[72] Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. 2025. A survey on lora of large language models. *Frontiers of Computer Science* 19, 7 (2025), 197605.

[73] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[74] Hanzi Mei, Dongqi Cai, Ao Zhou, Shangguang Wang, and Mengwei Xu. 2024. FedMoE: Personalized Federated Learning via Heterogeneous Mixture of Experts. *CoRR* abs/2408.11304 (2024). doi:10.48550/ARXIV.2408.11304 arXiv:2408.11304

[75] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 691–706.

[76] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. 2022. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8397–8406.

[77] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 739–753.

[78] Truc D. T. Nguyen, Phung Lai, Khang Tran, NhatHai Phan, and My T. Thai. 2023. Active Membership Inference Attack under Local Differential Privacy in Federated Learning. In *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain (Proceedings of Machine Learning Research, Vol. 206)*, Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent (Eds.). PMLR, 5714–5730. https://proceedings.mlr.press/v206/nguyen23e.html

[79] Kunjal Panchal, Nisarg Parikh, Sunav Choudhary, Lijun Zhang, Yuriy Brun, and Hui Guan. 2024. Thinking Forward: Memory-Efficient Federated Finetuning of Language Models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper_files/paper/2024/hash/7fc914993440219b64254e0c27964e11-Abstract-Conference.html

[80] Ivo Petrov, Dimitar I. Dimitrov, Maximilian Baader, Mark Niklas Müller, and Martin T. Vechev. 2024. DAGER: Exact Gradient Inversion for Large Language Models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper_files/paper/2024/hash/9ff1577a1f8308df1ccea6b4f64a103f-Abstract-Conference.html

[81] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty (Eds.). Association for Computational Linguistics, 487–503. doi:10.18653/V1/2021.EACL-MAIN.39

[82] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. 2017. Privacy-preserving deep learning: Revisited and enhanced. In *Applications and Techniques in Information Security: 8th International Conference, ATIS 2017, Auckland, New Zealand, July 6–7, 2017, Proceedings*. Springer, 100–110.

[83] Georg Pichler, Marco Romanelli, Leonardo Rey Vega, and Pablo Piantanida. 2023. Perfectly accurate membership inference by a dishonest central server in federated learning. *IEEE Transactions on Dependable and Secure Computing* (2023).

[84] Siqi Ping, Yuzhu Mao, Yang Liu, Xiao-Ping Zhang, and Wenbo Ding. 2024. FL-TAC: Enhanced Fine-Tuning in Federated Learning via Low-Rank, Task-Specific Adapter Clustering. *CoRR* abs/2404.15384 (2024). doi:10.48550/ARXIV.2404.15384 arXiv:2404.15384

[85] Jiaxing Qi, Zhongzhi Luan, Shaohan Huang, Carol J. Fung, Hailong Yang, and Depei Qian. 2024. FDLoRA: Personalized Federated Learning of Large Language Model via Dual LoRA Tuning. *CoRR* abs/2406.07925 (2024). doi:10.48550/ARXIV.2406.07925 arXiv:2406.07925

[86] Zhen Qin, Daoyuan Chen, Bingchen Qian, Bolin Ding, Yaliang Li, and Shuiguang Deng. 2024. Federated Full-Parameter Tuning of Billion-Sized Language Models with Communication Cost under 18 Kilobytes. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. https://openreview.net/forum?id=cit0hg4sEz

[87] Chen Qiu, Xingyu Li, Chaithanya Kumar Mummadi, Madan Ravi Ganesh, Zhenzhen Li, Lu Peng, and Wan-Yi Lin. 2023. Text-driven Prompt Generation for Vision-Language Models in Federated Learning. *CoRR* abs/2310.06123 (2023). doi:10.48550/ARXIV.2310.06123 arXiv:2310.06123

[88] Chen Qiu, Xingyu Li, Chaithanya Kumar Mummadi, Madan Ravi Ganesh, Zhenzhen Li, Lu Peng, and Wan-Yi Lin. 2024. Federated text-driven prompt generation for vision-language models. In *The Twelfth International Conference on Learning Representations*.

[89] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[90] Md. Rafi Ur Rashid, Vishnu Asutosh Dasu, Kang Gu, Najrin Sultana, and Shagufta Mehnaz. 2023. FLTrojan: Privacy Leakage Attacks against Federated Language Models Through Selective Weight Tampering. *CoRR* abs/2310.16152 (2023). doi:10.48550/ARXIV.2310.16152 arXiv:2310.16152

[91] Chao Ren, Han Yu, Hongyi Peng, Xiaoli Tang, Anran Li, Yulan Gao, Alysa Ziying Tan, Bo Zhao, Xiaoxiao Li, Zengxiang Li, and Qiang Yang. 2024. Advances and Open Challenges in Federated Learning with Foundation Models. *CoRR* abs/2404.15381 (2024). doi:10.48550/ARXIV.2404.15381 arXiv:2404.15381

[92] Yanli Ren, Mingqi Hu, Zhe Yang, Guorui Feng, and Xinpeng Zhang. 2024. BPFL: Blockchain-based privacy-preserving federated learning against poisoning attack. *Information Sciences* 665 (2024), 120377.

[93] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems* 31 (2018).

[94] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.

[95] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. 2022. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1354–1371.

[96] Jiyun Shin, Jinhyun Ahn, Honggu Kang, and Joonhyuk Kang. 2023. FedSplitX: Federated Split Learning for Computationally-Constrained Heterogeneous Clients. *CoRR* abs/2310.14579 (2023). doi:10.48550/ARXIV.2310.14579 arXiv:2310.14579

[97] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

[98] Abhishek Singh, Praneeth Vepakomma, Otkrist Gupta, and Ramesh Raskar. 2019. Detailed comparison of communication efficiency of split learning and federated learning. *CoRR* abs/1909.09145 (2019). arXiv:1909.09145 http://arxiv.org/abs/1909.09145

[99] Ningxin Su, Chenghao Hu, Baochun Li, and Bo Li. 2024. TITANIC: Towards production federated learning with large language models. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 611–620.

[100] Ningxin Su and Baochun Li. 2023. Asynchronous federated unlearning. In *IEEE INFOCOM 2023-IEEE conference on computer communications*. IEEE, 1–10.

[101] Shangchao Su, Bin Li, and Xiangyang Xue. 2025. Fedra: A random allocation strategy for federated tuning to unleash the power of heterogeneous clients. In *European Conference on Computer Vision*. Springer, 342–358.

[102] Yang Su, Na Yan, and Yansha Deng. 2024. Federated LLMs Fine-tuned with Adaptive Importance-Aware LoRA. *CoRR* abs/2411.06581 (2024). doi:10.48550/ARXIV.2411.06581 arXiv:2411.06581

[103] Guangyu Sun, Umar Khalid, Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, and Chen Chen. 2024. Conquering the Communication Constraints to Enable Large Pre-Trained Models in Federated Learning. arXiv:2210.01708 [cs.LG] https://arxiv.org/abs/2210.01708

[104] Rishub Tamirisa, Chulin Xie, Wenxuan Bao, Andy Zhou, Ron Arel, and Aviv Shamsian. 2024. FedSelect: Personalized Federated Learning with Customized Selection of Parameters for Fine-Tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[105] Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. 2022. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8485–8493.

[106] Yuanyishu Tian, Yao Wan, Lingjuan Lyu, Dezhong Yao, Hai Jin, and Lichao Sun. 2022. FedBERT: When federated learning meets pre-training. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 4 (2022), 1–26.

[107] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data poisoning attacks against federated learning systems. In *Computer security–ESORICs 2020: 25th European symposium on research in computer security, ESORICs 2020, guildford, UK, September 14–18, 2020, proceedings, part i 25*. Springer, 480–501.

[108] Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Position: Will we run out of data? Limits of LLM scaling based on human-generated data. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. https://openreview.net/forum?id=ViZcgDQjyG

[109] Minh Vu, Truc Nguyen, My T Thai, et al. 2024. Analysis of Privacy Leakage in Federated Large Language Models. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1423–1431.

[110] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris S. Papailiopoulos. 2020. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/b8ffa41d4e492f0fad2f13e29e1762eb-Abstract.html

[111] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. 2020. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems* 33 (2020), 16070–16084.

[112] Lin Wang, Zhichao Wang, and Xiaoying Tang. 2024. Save It All: Enabling Full Parameter Tuning for Federated Large Language Models via Cycle Block Gradient Descent. arXiv:2406.11187 [cs.LG] https://arxiv.org/abs/2406.11187

[113] Wenxuan Wang, Kuiyi Gao, Zihan Jia, Youliang Yuan, Jen-tse Huang, Qiuzhi Liu, Shuai Wang, Wenxiang Jiao, and Zhaopeng Tu. 2024. Chain-of-Jailbreak Attack for Image Generation Models via Editing Step by Step. *CoRR* abs/2410.03869 (2024). doi:10.48550/ARXIV.2410.03869 arXiv:2410.03869

[114] Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. 2024. A Survey on Efficient Federated Learning Methods for Foundation Model Training. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*. ijcai.org, 8317–8325. https://www.ijcai.org/proceedings/2024/919

[115] Chen Wu, Xi Li, and Jiaqi Wang. 2024. Vulnerabilities of Foundation Model Integrated Federated Learning Under Adversarial Threats. *CoRR* abs/2401.10375 (2024). doi:10.48550/ARXIV.2401.10375 arXiv:2401.10375

[116] Feijie Wu, Zitao Li, Yaliang Li, Bolin Ding, and Jing Gao. 2024. Fedbiot: Llm local fine-tuning in federated learning without full model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3345–3355.

[117] Geming Xia, Jian Chen, Chaodong Yu, and Jun Ma. 2023. Poisoning Attacks in Federated Learning: A Survey. *IEEE Access* (2023), 10708–10722. Issue 2.

[118] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. 2019. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*.

[119] Jie Xu, Karthikeyan Saravanan, Rogier van Dalen, Haaris Mehmood, David Tuckey, and Mete Ozay. 2024. DP-DyLoRA: Fine-Tuning Transformer-Based Models On-Device under Differentially Private Federated Learning using Dynamic Low-Rank Adaptation. *CoRR* abs/2405.06368 (2024). doi:10.48550/ARXIV.2405.06368 arXiv:2405.06368

[120] Mengwei Xu, Dongqi Cai, Yaozong Wu, Xiang Li, and Shangguang Wang. 2024. FwdLLM: Efficient federated finetuning of large language models with perturbed inferences. In *USENIX ATC*.

[121] Zhao Xu, Fan Liu, and Hao Liu. 2024. Bag of Tricks: Benchmarking of Jailbreak Attacks on LLMs. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir

Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper_files/paper/2024/hash/38c1dfb4f7625907b15e9515365e7803-Abstract-Datasets_and_Benchmarks_Track.html

[122] Zheng Xu, Yanxiang Zhang, Galen Andrew, Christopher A. Choquette-Choo, Peter Kairouz, H. Brendan McMahan, Jesse Rosenstock, and Yuanbo Zhang. 2023. Federated Learning of Gboard Language Models with Differential Privacy. In *Proceedings of the The 61st Annual Meeting of the Association for Computational Linguistics: Industry Track, ACL 2023, Toronto, Canada, July 9-14, 2023*, Sunayana Sitaram, Beata Beigman Klebanov, and Jason D. Williams (Eds.). Association for Computational Linguistics, 629–639. doi:10.18653/V1/2023.ACL-INDUSTRY.60

[123] Haomiao Yang, Mengyu Ge, Dongyun Xue, Kunlan Xiang, Hongwei Li, and Rongxing Lu. 2023. Gradient leakage attacks in federated learning: Research frontiers, taxonomy and future directions. *IEEE Network* (2023).

[124] Tien-Ju Yang, Yonghui Xiao, Giovanni Motta, Françoise Beaufays, Rajiv Mathews, and Mingqing Chen. 2023. Online Model Compression for Federated Learning with Large Models. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1–5. doi:10.1109/ICASSP49357.2023.10097124

[125] Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. 2021. Be Careful about Poisoned Word Embeddings: Exploring the Vulnerability of the Embedding Layers in NLP Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, 2048–2058. doi:10.18653/V1/2021.NAACL-MAIN.165

[126] Yiyuan Yang, Guodong Long, Tao Shen, Jing Jiang, and Michael Blumenstein. 2024. Dual-Personalizing Adapter for Federated Foundation Models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper_files/paper/2024/hash/45a30141c6719e9cfedfb51f1c665a37-Abstract-Conference.html

[127] Zheng Yang, Ke Gu, and Yiming Zuo. 2024. Byzantine Robust Federated Learning Scheme Based on Backdoor Triggers. *Computers Materials&Continua* (2024), 2813–2831. Issue 5.

[128] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing* (2024), 100211.

[129] Yuhang Yao, Jianyi Zhang, Junda Wu, Chengkai Huang, Yu Xia, Tong Yu, Ruiyi Zhang, Sungchul Kim, Ryan A. Rossi, Ang Li, Lina Yao, Julian J. McAuley, Yiran Chen, and Carlee Joe-Wong. 2024. Federated Large Language Models: Current Progress and Future Directions. *CoRR* abs/2409.15723 (2024). doi:10.48550/ARXIV.2409.15723 arXiv:2409.15723

[130] Rui Ye, Rui Ge, Fengting Yuchi, Jingyi Chai, Yanfeng Wang, and Siheng Chen. 2024. Leveraging unstructured text data for federated instruction tuning of large language models. In *International Workshop on Trustworthy Federated Learning*. Springer, 119–131.

[131] Liping Yi, Han Yu, Gang Wang, and Xiaoguang Liu. 2023. FedLoRA: Model-Heterogeneous Personalized Federated Learning with LoRA Tuning. *CoRR* abs/2310.13283 (2023). doi:10.48550/ARXIV.2310.13283 arXiv:2310.13283

[132] yiyuan yang, Guodong Long, Tianyi Zhou, Qinghua Lu, Shanshan Ye, and Jing Jiang. 2025. Federated Adapter on Foundation Models: An Out-Of-Distribution Approach. https://openreview.net/forum?id=LcpdPCkZwI

[133] KiYoon Yoo and Nojun Kwak. 2022. Backdoor Attacks in Federated Learning by Rare Embeddings and Gradient Ensembling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 72–88. doi:10.18653/V1/2022.EMNLP-MAIN.6

[134] Sixing Yu, J. Pablo Muñoz, and Ali Jannesari. 2023. Bridging the Gap Between Foundation Models and Heterogeneous Federated Learning. *CoRR* abs/2310.00247 (2023). doi:10.48550/ARXIV.2310.00247 arXiv:2310.00247

[135] Sixing Yu, Juan Pablo Muñoz, and Ali Jannesari. 2024. Federated Foundation Models: Privacy-Preserving and Collaborative Learning for Large Models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, Nicoletta Calzolari, Min-Yen Kan, Véronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (Eds.). ELRA and ICCL, 7174–7184. https://aclanthology.org/2024.lrec-main.630

[136] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 1–9. doi:10.18653/V1/2022.ACL-SHORT.1

[137] Oualid Zari, Chuan Xu, and Giovanni Neglia. 2021. Efficient passive membership inference attack in federated learning. *CoRR* abs/2111.00430 (2021). arXiv:2111.00430 https://arxiv.org/abs/2111.00430

[138] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the Catastrophic Forgetting in Multimodal Large Language Models. *CoRR* abs/2309.10313 (2023). doi:10.48550/ARXIV.2309.10313 arXiv:2309.10313

[139] Chunxu Zhang, Guodong Long, Hongkuan Guo, Xiao Fang, Yang Song, Zhaojie Liu, Guorui Zhou, Zijian Zhang, Yang Liu, and Bo Yang. 2024. Federated Adaptation for Foundation Model-based Recommendations. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*. ijcai.org, 5453–5461. https://www.ijcai.org/proceedings/2024/603

[140] Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, and Shui Yu. 2020. PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet of Things Journal* 8, 5 (2020), 3310–3322.

[141] Jianxin Zhang, Mengda Zhao, Zhenwei Wang, Weijian Su, and Pengfei Wang. 2025. Model Recovery in Federated Unlearning With Restricted Server Data Resources. *IEEE Internet of Things Journal* (2025).

[142] Liwei Zhang, Linghui Li, Xiaoyong Li, Binsi Cai, Yali Gao, Ruobin Dou, and Luying Chen. 2023. Efficient Membership Inference Attacks against Federated Learning via Bias Differences. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*. 222–235.

[143] Zeling Zhang, Dongqi Cai, Yiran Zhang, Mengwei Xu, Shangguang Wang, and Ao Zhou. 2024. FedRDMA: Communication-Efficient Cross-Silo Federated LLM via Chunked RDMA Transmission. In *Proceedings of the 4th Workshop on Machine Learning and Systems, EuroMLSys 2024, Athens, Greece, 22 April 2024*. ACM, 126–133. doi:10.1145/3642970.3655834

[144] Zhiyuan Zhang, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Fed-FA: theoretically modeling client data divergence for federated language backdoor defense. *Advances in Neural Information Processing Systems* 36 (2023), 62006–62031.

[145] Zhuo Zhang, Xiangjing Hu, Jingyuan Zhang, Yating Zhang, Hui Wang, Lizhen Qu, and Zenglin Xu. 2023. Fedlegal: The first real-world federated learning benchmark for legal nlp. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 3492–3507.

[146] Zixin Zhang, Fan Qi, and Changsheng Xu. [n. d.]. Enhancing Storage and Computational Efficiency in Federated Multimodal Learning for Large-Scale Models. In *Forty-first International Conference on Machine Learning*.

[147] Zhiyuan Zhang, Qi Su, and Xu Sun. 2022. Dim-Krum: Backdoor-Resistant Federated Learning for NLP with Dimension-wise Krum-Based Aggregation. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 339–354. doi:10.18653/V1/2022.FINDINGS-EMNLP.25

[148] Zikai Zhang, Jiahao Xu, Ping Liu, and Rui Hu. 2024. Fed-piLot: Optimizing LoRA Assignment for Efficient Federated Foundation Model Fine-Tuning. *CoRR* abs/2410.10200 (2024). doi:10.48550/ARXIV.2410.10200 arXiv:2410.10200

[149] Jujia Zhao, Wenjie Wang, Chen Xu, Zhaochun Ren, See-Kiong Ng, and Tat-Seng Chua. 2024. LLM-based Federated Recommendation. *CoRR* abs/2402.09959 (2024). doi:10.48550/ARXIV.2402.09959 arXiv:2402.09959

[150] Zihao Zhao, Zhenpeng Shi, Yang Liu, and Wenbo Ding. 2023. Inclusive Data Representation in Federated Learning: A Novel Approach Integrating Textual and Visual Prompt. In *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing*. 724–729.

[151] Fei Zheng. 2023. Input Reconstruction Attack against Vertical Federated Large Language Models. *CoRR* abs/2311.07585 (2023). doi:10.48550/ARXIV.2311.07585 arXiv:2311.07585

[152] Jiaying Zheng, Hainan Zhang, Lingxiang Wang, Wangjie Qiu, Hong-Wei Zheng, and Zhi Ming Zheng. 2024. Safely Learning with Private Data: A Federated Learning Framework for Large Language Model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, 5293–5306. https://aclanthology.org/2024.emnlp-main.303

[153] Zan Zhou, Changqiao Xu, Bo Wang, Tengfei Li, Sizhe Huang, Shujie Yang, and Su Yao. 2024. SecFFT: Safeguarding Federated Fine-Tuning for Large Vision Language Models against Covert Backdoor Attacks in IoRT Networks. *IEEE Internet of Things Journal* (2024).

[154] Didi Zhu, Zhongyi Sun, Zexi Li, Tao Shen, Ke Yan, Shouhong Ding, Chao Wu, and Kun Kuang. 2024. Model Tailor: Mitigating Catastrophic Forgetting in Multi-modal Large Language Models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. https://openreview.net/forum?id=piujJIF3zs

[155] Junyi Zhu and Matthew B. Blaschko. 2021. R-GAP: Recursive Gradient Attack on Privacy. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=RSU17UoKfJF

[156] Jianhao Zhu, Changze Lv, Xiaohua Wang, Muling Wu, Wenhao Liu, Tianlong Li, Zixuan Ling, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2024. Promoting Data and Model Privacy in Federated Learning through Quantized LoRA. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, 10501–10512. https://aclanthology.org/2024.findings-emnlp.615

[157] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in neural information processing systems* 32 (2019).

[158] Xiangrong Zhu, Guangyao Li, and Wei Hu. 2023. Heterogeneous federated knowledge graph embedding learning and unlearning. In *Proceedings of the ACM web conference 2023*. 2444–2454.

[159] Weiming Zhuang, Chen Chen, and Lingjuan Lyu. 2023. When Foundation Model Meets Federated Learning: Motivations, Challenges, and Future Directions. *CoRR* abs/2306.15546 (2023). doi:10.48550/ARXIV.2306.15546 arXiv:2306.15546

[160] Xuhan Zuo, Minghao Wang, Tianqing Zhu, Lefeng Zhang, Dayong Ye, Shui Yu, and Wanlei Zhou. 2024. Federated TrustChain: Blockchain-Enhanced LLM Training and Unlearning. *CoRR* abs/2406.04076 (2024). doi:10.48550/ARXIV.2406.04076 arXiv:2406.04076

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009