https://doi.org/10.1007/s11704-025-50480-3

REVIEW ARTICLE

Federated reasoning LLMs: a survey

Shuyue WEI[1], Yongxin TONG[1]✉, Zimu ZHOU[2]✉, Yi XU[1]✉, Jingkai GAO[1], Tongyu WEI[1], Tianran HE[1], Weifeng LV[1]✉

1. State Key Laboratory of Complex & Critical Software Environment, Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, School of Computer Science and Engineering, Beihang University, Beijing 100191, China
2. Department of Data Science, City University of Hong Kong, Hong Kong 999077, China

Received April 18, 2025; accepted April 29, 2025

E-mail: yxtong@buaa.edu.cn; zimuzhou@cityu.edu.hk; xuy@buaa.edu.cn; lwf@buaa.edu.cn

Abstract

Reasoning has long been regarded as a distinctive hallmark of human cognition, and recent advances in the artificial intelligence community have increasingly focused on the reasoning large language models (`rLLMs`). However, due to strict privacy regulations, the domain-specific reasoning knowledge is often distributed across multiple data owners, limiting the `rLLM`'s ability to fully leverage such valuable resources. In this context, federated learning (FL) has gained increasing attention in both the academia and industry as a promising privacy-preserving paradigm for addressing the challenges in the data-efficient training of `rLLMs`.

In this paper, we conduct a comprehensive survey on federated `rLLMs` and propose a novel taxonomy based on training signals, including training signals derived from raw data, learned representations, and preference feedback. For each category, we emphasize the emerging trends according to how to use FL to enhance reasoning capabilities of `rLLMs` considering the model effectiveness, communication cost and privacy preservation. Finally, we envision future research directions and challenges based on insights from existing studies.

Keywords

federated learning; reasoning LLMs; fine tuning; retrieval-augmented generation

1 Introduction

*"Man is the only animal capable of **reasoning**, though many others possess the faculty of memory and instruction in common with him."*

—Aristotle

Associated with characteristically human activities, reasoning has long been regarded as the distinguishing capability of humans [1]. Specifically, as defined in the Cambridge Dictionary [2], the term **reasoning** refers to the process of thinking about something in order to make a decision. For example, in classical deductive reasoning [3], given two propositions $\mathbf{A} = \textbf{True}$ and $\mathbf{A} \to \mathbf{B}$, we can make the deduction to obtain the final answer as follows,

$$\underbrace{If\ \mathbf{A} = \textbf{True} \text{ and } \mathbf{A} \to \mathbf{B}}_{\text{Reasoning Process}}, \text{ then } \underbrace{\mathbf{B} = \text{ True}}_{\text{Final Answer}}. \tag{1}$$

Reasoning is widely regarded as a fundamental capability that large language models (`LLMs`) must develop on the path toward the Artificial General Intelligence (AGI) [4–7]. Recently, enhancing the reasoning capabilities of `LLMs` has emerged as a central research direction in both academia [8–10] and industry [11–14]. For example, the technical reports of advanced LLMs released in 2025, such as DeepSeek-R1 [11], GPT-4.5 [12], Claude 3.75 [13], and QwQ-32B [14], highlight reasoning as both a central goal and a major achievement.

Notably, in the seminal work, Wei et al. [8] introduced Chain-of-Thought (CoT) as a mechanism to facilitate reasoning in `LLMs`, which enables the language model to "think more" through multiple intermediate steps before generating a final output. Given the remarkable effectiveness of CoT-style reasoning in complex tasks such as code generation and mathematical problem solving, the prevailing paradigm of the language model is increasingly shifting toward the reasoning-oriented LLMs (`rLLMs`).

In the training process of `rLLMs`, high-quality data, particularly those with explicit or implicit reasoning paths, serve as the cornerstone. However, in real-world applications, the valuable domain-specific knowledge, e.g., clinical decision records in healthcare [15] or proprietary codes in soft engineering [16], is typically distributed across multiple data owners due to the strict privacy regulations [17]. Therefore, how to utilize these reasoning-rich training signals collected from human experts or generated by LLMs in a privacy-preserving fashion is crucial to enhance reasoning capabilities of `rLLMs`. As a remedy, federated learning (FL) [18–23], as a novel distributed learning paradigm, enables `rLLMs` trained on domain-specific reasoning data across data owners while preserving privacy and complying with constraints in data regulation.

Different from previous smaller LMs, the rLLM's unique features (e.g., billion-scale parameters, in-context learning capabilities) bring new challenges and opportunities for federated learning in accuracy, privacy and communication, which largely reshape traditional FL approaches. Researchers have developed numerous novel FL-based approaches to improve rLLMs, including federated pre-training [24–33], federated instruction tuning [34–38], federated prompt learning [39–48], etc. Therefore, it is beneficial to conduct a systematic survey in this rapidly developing field, which can inspire future research on FL of rLLMs by providing evolving landscapes and understanding unique challenges and promising opportunities along this direction.

Related surveys on FL of LLMs. Existing surveys [49–53], which investigate the federated LLMs, mainly focus on FL techniques using training signals from raw data or representations, overlooking the flexible training signals for rLLMs. In contrast, this survey paper provides a more comprehensive survey covering used training signals from raw data to preferences. We compare FL techniques of rLLMs covered in this survey and the previous in Table 1.

Main contributions. In this paper, we presents a comprehensive survey on the federated learning of reasoning LLMs. Specifically, the main contributions of this survey are summarized as follows:

Firstly, to accommodate the shift from prior small language models to large language models, we propose a novel taxonomy for federated rLLMs, organized according to forms of training signals. Specifically, we categorize training signals fed into rLLMs into three types: i.e., (i) signals derived from raw data, (ii) model-interpretable representations and (iii) preferences from human or AI models.

Secondly, for each class of FL techniques, we discuss the challenges or opportunities introduced by the features of rLLMs concerning model accuracy, communication overhead, and privacy preservation. We also summarize emerging research trends and highlight promising directions for future research.

Lastly, we envision two future research avenues that are particularly promising for enhancing reasoning capabilities in federated scenarios, both remaining in early stages: (i) Federated RL enhanced rLLMs and (ii) Federated RAG enhanced rLLMs.

Survey structure and roadmap. This survey provides a comprehensive overview for the federated training techniques, open-source platforms, typical applications and future directions for rLLMs. As shown in Fig. 1, this survey is organized as follows. We firstly introduce our proposed new taxonomy for FL of rLLMs based on training signals in Section 3. Section 4 presents the FL techniques based on training signals from distributed raw data, which mainly utilizes the federated supervised learning and includes the federated supervised pre-training in Section 4.1 and federated instruction tuning in Section 4.2. Section 5 discusses model-interpretable training signals (i.e., learned representation), including federated prompt learning in Section 5.1, federated adapter learning in Section 5.2 and the federated knowledge distillation in Section 5.3. We then review the studies utilizing preference signals from human or AI models, which primarily employ the federated reinforcement learning in Section 6. Next, we review the open-source platforms for federated rLLMs and the representative applications in Section 7. Finally, we envision the future research directions in Section 8 and conclude this work in Section 9.

2 Concepts: reasoning LLMs and FL

In this section, we briefly review large language models (LLMs) and compare the two response modes: straightforward question-answering mode and multi-step reasoning mode. We then introduce fundamental concepts of federated learning (FL) and discuss commonly used privacy-preserving techniques.

2.1 Language models and its reasoning

Language model (LM). LMs [81] define the probability distribution over a sequence of tokens. Given a sequence of tokens $X=(x_1,x_2,\ldots,x_n)$, the LMs model their joint probability and support a series of of fundamental natural language processing (NLP) tasks such as text summarization [82], machine translation [83], and text completion [84] through next-token prediction. The next-token prediction process can be typically formulated as follows,

Table 1 Related surveys on federated reasoning large language models

Year		① Signal from raw data		② Signal from representation			③ Signal from preference	
		Pre-training	Instruct.-tuning	Prompt-tuning	Adapter-tuning	Know.-distil.	Human-preference	AI-preference
2023	[49]	√	√	√				
2024	[54]			√	√	√		
2024	[55]	√	√	√	√	√		
2024	[50]		√	√	√	√		
2024	[53]	√	√	√	√		√	
2025	[51]	√	√	√	√	√		
2025	[52]		√	√	√	√		
This survey		√	√	√	√	√	√	√

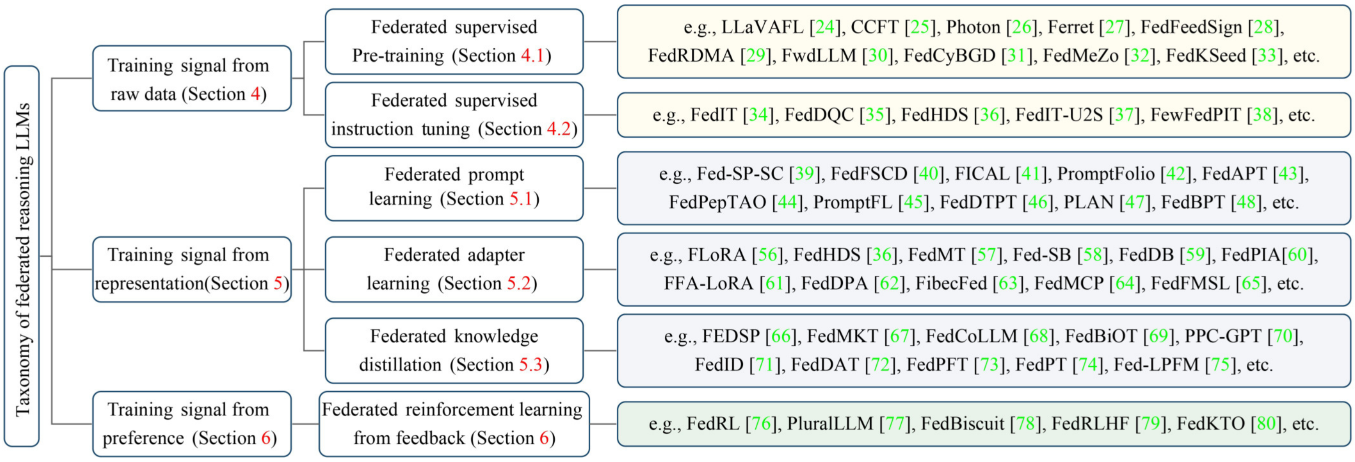

Taxonomy of federated reasoning LLMs
Training signal from raw data (Section 4)
Federated supervised Pre-training (Section 4.1)
e.g., LLaVAFL [24], CCFT [25], Photon [26], Ferret [27], FedFeedSign [28], FedRDMA [29], FwdLLM [30], FedCyBGD [31], FedMeZo [32], FedKSeed [33], etc.
Federated supervised instruction tuning (Section 4.2)
e.g., FedIT [34], FedDQC [35], FedHDS [36], FedIT-U2S [37], FewFedPIT [38], etc.
Training signal from representation(Section 5)
Federated prompt learning (Section 5.1)
e.g., Fed-SP-SC [39], FedFSCD [40], FICAL [41], PromptFolio [42], FedAPT [43], FedPepTAO [44], PromptFL [45], FedDTPT [46], PLAN [47], FedBPT [48], etc.
Federated adapter learning (Section 5.2)
e.g., FLoRA [56], FedHDS [36], FedMT [57], Fed-SB [58], FedDB [59], FedPIA[60], FFA-LoRA [61], FedDPA [62], FibecFed [63], FedMCP [64], FedFMSL [65], etc.
Federated knowledge distillation (Section 5.3)
e.g., FEDSP [66], FedMKT [67], FedCoLLM [68], FedBiOT [69], PPC-GPT [70], FedID [71], FedDAT [72], FedPFT [73], FedPT [74], Fed-LPFM [75], etc.
Training signal from preference (Section 6)
Federated reinforcement learning from feedback (Section 6)
e.g., FedRL [76], PluralLLM [77], FedBiscuit [78], FedRLHF [79], FedKTO [80], etc.

Fig. 1 A taxonomy for federated reasoning LLMs based on the nature of training signals

$$P(y \mid X) = \prod_{t=1}^{T} P(y_t \mid X, y_1, y_2, \ldots, y_{t-1}), \tag{2}$$

where X is the input context sequence and $Y = (y_1, y_2, \ldots, y_T)$ is the target predicted sequence.

Large language model (LLM). In recent years, we have witnessed remarkable progress in LLMs. With the scaling of model parameters and training data, LLMs (e.g., GPT-4 [85], PaLM-2 [86], and LLaMA-2 [87]) have demonstrated exceptional performance across a wide range of NLP tasks, surpassing the traditional LM, which is referred to the *Emergent Abilities* of LLMs [88]. For example, LLM first exhibits in-context learning abilities, enabling them to perform new tasks by following a few examples provided in the input context without relying on gradient-based parameter optimization. Thus, LLMs are also viewed as zero-shot learners [89]. At the same time, LLMs show emerging reasoning capabilities, particularly in solving complex tasks requiring multiple steps of processing, such as mathematical computations, code generation, and logical inference [4].

This survey focuses on reasoning-oriented large language models (rLLMs), emphasizing the multi-step reasoning capability in complex tasks. Next, we introduce the multiple-step reasoning of rLLMs from the perspective of Chain-of-Thoughts (CoT) [8].

Reasoning LLM (rLLM). The *Chain of Thought has emerged as a novel reasoning paradigm for* LLM [90–92], attracting significant attention from the researchers in both academia and industry [10,86,93–96]. In the seminal work [8], Wei et al. have observed a crucial insight in the reasoning process of LLMs, i.e., if we allow the LLMs to think or reason step-by-step before arriving at the final answer, the LLM's performances on complex tasks such as math word problems from GSM8K [97] can be significantly enhanced. As shown in Fig. 2, the CoT reasoning paradigm allows the LLMs to generate reasoning tokens before output the answer sequence. Similarly, we can formalize the CoT reasoning process through next-token prediction as,

$$\begin{aligned} P(y \mid X) &= \prod_{t=1}^{T_r} P(y_t \mid X, y_1, y_2, \ldots, y_{t-1}) \\ &= \underbrace{\prod_{t=1}^{k} P(r_t \mid X, r_{<t})}_{\textbf{Reasoning Tokens}} \cdot \underbrace{\prod_{t=k+1}^{k+T_r} P(y_{t-k} \mid X, r_{<k}, y_{<t})}_{\textbf{Answer Tokens}}, \end{aligned} \tag{3}$$

where $X = (x_1, x_2, \ldots, x_n)$ is the given context tokens, the reasoning tokens $R = (r_1, r_2, \ldots, r_k)$ forming the CoT reasoning path, and $Y = (y_1, y_2, \ldots, y_{T_r})$ is the final answer to the user's query. In an explicit CoT reasoning process, both the reasoning path R and the final answer Y are generated and presented to the user by the reasoning LLMs. Compared with the standard LLMs that respond to the user queries straightforwardly, rLLMs not only enhance the capability to handle complex tasks but also improve interpretability by generating explicit reasoning tokens during the inference process. We further illustrate the token generation process of standard LLMs and reasoning LLMs through a toy example.

Example 1 As illustrated in Fig. 3, consider the following scenario: a LLM user poses a mathematic question, i.e., *"If Person A has 5 apples, Person B has 3 more apples than Person A, and Person C has twice as many apples as Person B, how many apples does C have?"* (i) Straightforward Query: The LLM attempts to answer the question directly without performing intermediate reasoning steps. It incorrectly output 13 apples for Person C, possibly due to flawed

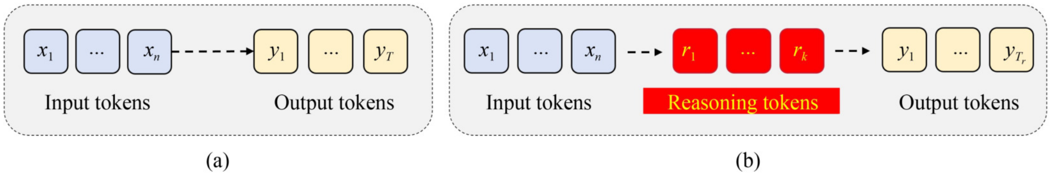
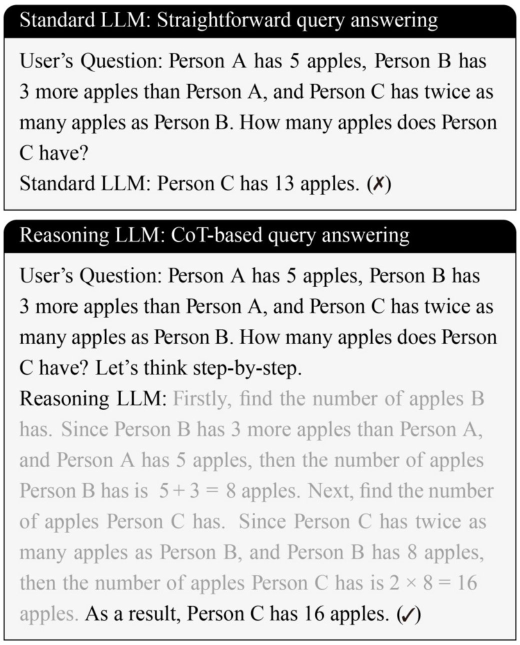


Fig. 2 The next-token-prediction in the Standard LLMs (a) and Reasoning LLMs (b)



Standard LLM: Straightforward query answering

User’s Question: Person A has 5 apples, Person B has 3 more apples than Person A, and Person C has twice as many apples as Person B. How many apples does Person C have?
Standard LLM: Person C has 13 apples. (✗)

Reasoning LLM: CoT-based query answering

User’s Question: Person A has 5 apples, Person B has 3 more apples than Person A, and Person C has twice as many apples as Person B. How many apples does Person C have? Let’s think step-by-step.
Reasoning LLM: Firstly, find the number of apples B has. Since Person B has 3 more apples than Person A, and Person A has 5 apples, then the number of apples Person B has is $5+3=8$ apples. Next, find the number of apples Person C has. Since Person C has twice as many apples as Person B, and Person B has 8 apples, then the number of apples Person C has is $2\times 8=16$ apples. As a result, Person C has 16 apples. (✓)

Fig. 3 Two types response modes in LLMs: straightforward query answering (top) and multi-steps reasoning (bottom)

arithmetic such as misinterpreting the expression $5+3\times 2$ as 11, or other erroneous combinations. (ii) Chain-of-Thought Query: The LLM follows a step-by-step reasoning process: it first calculates the Person B’s apples as $5+3=8$, then computes Person C’s apples as twice that amount, $2\times 8=16$. Taking CoT-based reasoning enables LLM to arrive at correct answer through an interpretable and logically grounded process.

The training of `rLLMs` requires a data-efficient approach heavily relying on vast amounts of publicly available data from the internet, however, the domain-specific data of many real-world scenarios (e.g., healthcare [98–100] and software engineering [101]), remains limited and often inaccessible due to privacy and ownership constraints. As shown in Fig. 4, it is estimated that only less than 10% of surface web data is publicly accessible, while the remaining 90% resides in the access-controlled deep web [102]. Due to stringent privacy regulations such as the GDPR [17] and CCPA [103], data owners (e.g., institutions or companies) are restricted from directly sharing their valuable and sensitive data. This leads to the phenomenon called data silos, i.e., large volumes of valuable datasets remain isolated and can not be fully leveraged by today’s data-hungry machine learning models. In this context, federated learning, a distributed training paradigm without compromising data privacy, has emerged as a crucial technique to unlock the reasoning potential of `rLLMs`. Therefore, if we aim to harness private, domain-specific valuable data from the deep web, FL can also serve as a privacy-friendly learning paradigm and *remains alive* to AI community [104]. Next, we briefly introduce the background and basic concepts of FL and review the commonly used privacy-preserving techniques.

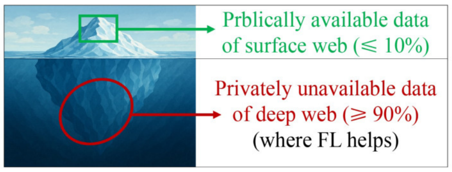

Prblically available data of surface web (≤ 10%)
Privately unavailable data of deep web (≥ 90%) (where FL helps)

Fig. 4 The Data across both the surface web (freely indexed) and the deep web (non-indexed or access-controlled)

2.2 Federated learning

This subsection first introduces the fundamental concepts of federated learning and then presents a general FL framework. At last, we discuss widely used privacy-preserving techniques in FL scenarios, including differential privacy, homomorphic encryption, and secure multi-party computation.

Federated learning (FL). As a response to data privacy regulations, federated learning (FL) [18–20] has emerged as a novel learning paradigm that enables an easy use of distributed datasets across data owners without compromising the privacy. For early FL algorithms (e.g., FedAvg [105]), FL clients updated the global model through the shared local gradients. In recent years, researchers have explored various alternative formats of local knowledge to shared across FL clients, such as logits [45,67,100] or small proxy models [73,75,106]. To this end, we introduce a general FL framework in this survey to encompass the various formats of knowledge exchanged across the FL clients during the jointly training. The general FL framework is illustrated in Algorithm 1, which outlines the procedures executed at the FL server and the FL clients.

- (i) Acts at FL server: In each training rounds the FL server will receive knowledge shared by a set of FL clients C (in line 4). Then, in line 5, the FL server aggregates all received local knowledge K_c from participating clients. The server aggregated all local knowledge through the aggregation function $\mathcal{G}$.**aggregation**. Finally, the FL server prepare through $\mathcal{G}$.**prepare** and then send the global knowledge K_G for participating FL clients.

- (ii) Acts at FL client: In lines 9−10, the FL client c first downloads the global knowledge K_G and updates its local model $\mathcal{L}$ using

```
Algorithm 1 General FL framework
-------- Acts at FL server $S$ --------:
1  if $\mathcal{G}$.init = $false$ then
2      Initialize the global model $\mathcal{G}$;
3  else
4      Receive local knowledge $\{K_c|c \in C\}$;
5      $\mathcal{A}_G = \mathcal{G}$.aggregation $(\mathcal{G}, \{K_c|c \in C\})$;
6      $\mathcal{G}$.update $(\mathcal{A}_G, D_G\})$;
7  Prepare the global knowledge for FL client
       $K_G \leftarrow \mathcal{G}$.prepare();
8  Send global knowledge $K_G$ to FL clients;
-------- Acts at FL client $c$- -------:
9  Download $K_G$ from FL server;
10 Update local model $\mathcal{L}$.update $(D_c, K_G)$ ;
11 Prepare the shared local knowledge
       $K_c \leftarrow \mathcal{L}$.prepare();
12 Send local knowledge $K_c$ to the server;
```

$\mathcal{L}$**.update** based on its private dataset D_c in combination with the received global knowledge K_G. Then, the FL client prepares its local knowledge K_c using $\mathcal{L}$**.prepare** for aggregation and sends it back to the FL server.

The above actions and interactions between the FL server and the participating FL clients are performed iteratively until the FL training process converges or meets a predefined stopping criterion.

Privacy-preserving techniques for FL. In the following, we review commonly used privacy techniques in FL setups, including: (i) differential privacy [107,108], (ii) secure multi-party computation [110], and (iii) homomorphic encryption [109].

- Differential privacy (DP): Proposed by Dwork [107], it provides formal guarantees by injecting calibrated noise into the private value. Formally, we refer to a randomized mechanism $\mathcal{M}$ satisfying (ϵ,δ)-DP, if for all measurable subsets $\mathcal{S}$ in output space and for all neighboring datasets D and D' (differing in at most one element), formally as,

$$\Pr[\mathcal{M}(D)\in\mathcal{S}] \leqslant e^{\epsilon}\cdot\Pr[\mathcal{M}(D')\in\mathcal{S}]+\delta, \tag{4}$$

where ϵ and δ denotes the privacy budgets. In the context of FL, the DP is typically implemented by adding noise to the local gradients before they are shared with the FL server to make aggregation.

- Secure multi-party computation (SMC): The concept of SMC was first introduced by Yao [111], enabling multiple parties to jointly compute arbitrary functions over their private inputs without revealing those inputs to one another [110]. In particular, *secure aggregation* [105], a class of SMC protocols tailored for operations such as SUM or MULT, enhances the computational efficiency and supports scalability to a larger number of participants.

- Homomorphic encryption (HE): The HE [109] allows computation to be performed directly on encrypted data just like that on the plaintext. In 2009, Gentry [109] proposed the fully homomorphic encryption (Fully-HE) that supports both the SUM and MULT on the ciphertext, particularly suitable for use in FL [19]. Specifically, FL clients can encrypt their local updates and transmit them to the FL server, which can perform aggregation directly on encrypted data without the need for decryption.

Summary of privacy techniques. We summarize the features of the representative privacy-preserving techniques commonly used in FL scenarios from five dimensions in Table 2, including *privacy protection level, accuracy, computation cost, communication cost, and supported operations*. As no single privacy-preserving technique universally outperforms others across all dimensions, i.e., there is no free lunch in privacy preservation [112]. Thus, it is crucial to select an appropriate method based on the specific requirements and constraints of application scenarios. For example, a recent study [113] in FL demonstrates that federated graph learning (e.g., ASTGCN [114]) in the aggregation phase relies solely on the summation operation, making it well-suited for efficient and secure training via *secure aggregation*.

3 Overview and taxonomy

This section introduces prior FL approaches for language models. We begin by comparing FL solutions across traditional language models (LMs), large language models (LLMs), and reasoning large language models (rLLMs). We then present a novel taxonomy of federated training approaches based on the nature of training signals, which can be categorized into three levels, i.e., (i) raw data level, (ii) model-interpretable representation level, and (iii) human or AI preference level. Finally, we review representative prior studies corresponding to each level of training signal in more detail.

As aforementioned, the emergent and reasoning abilities of LLMs/rLLMs enable these models to incorporate a broader range of training signals, which, in turn, opens up new opportunities to use diverse federated training approaches to enhance themselves. Specifically, we summarize and compare the FL approaches for LMs, LLMs, and rLLMs in Table 3.

Table 2 Comparison of the representative privacy-preserving techniques for FL (more ⋆ denote better performance)

Techniques	Privacy-level	Accuracy	Computation cost	Commutation cost	Supported operations
DP [107,108]	⋆	⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆
Fully-HE [109]	⋆⋆⋆	⋆⋆⋆	⋆⋆	⋆⋆	⋆⋆
General SMC [110]	⋆⋆⋆	⋆⋆⋆	⋆	⋆	⋆⋆⋆
Secure Agg. [105]	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆	⋆

Table 3 Federated learning of different language models, including LMs, LLMs and rLLMs (*a.k.a.* Reasoning LLMs)

	① Signal from raw data		② Signal from representation			③ Signal from preference	
	Pre-training	Instruct.-tuning	Prompt-tuning	Adapter-tuning	Know.-distil.	Human-preference	AI-preference
LMs	√			√	√		
LLMs	√	√	√	√	√	√	
rLLMs	√	√	√	√	√	√	√

Evolution of training signals. Firstly, a central research topic of conventional deep learning-based `LMs` is to leverage large-scale datasets to pre-train models for a wide range of downstream tasks. The training processes are typically driven by signals inherent in the raw data, employing supervised or self-supervised learning approaches [115–117]. Additionally, transfer learning techniques (e.g., adapter approach [118] or knowledge distillation [119]) are commonly employed to enhance model performance further. Secondly, `LLMs` are widely regarded as few-shot or zero-shot learners [89,120], allowing them to learn from more abstract training signals, such as prompts or proxy models, which has given rise to a suite of novel techniques designed for `LLMs`, such as instruction tuning and prompt learning [5,7]. These further improve the `LLM`’s ability to understand and follow user instructions during query-answering interactions. Finally, in the era of `rLLMs`, researchers have not only extended the use of the aforementioned techniques but have also begun to incorporate preference-level signals sourced from either humans or AI as the training signals to further enhance the reasoning capabilities of language models [11,85].

Taxonomy. Similarly, when FL meets the `rLLMs`, we can use the same taxonomy for federated training techniques designed to enhance the reasoning capabilities of `rLLMs` based on the different forms of training signals. Note that, these categories naturally align with the three major FL paradigms, i.e., (i) federated supervised learning (Fed-ST) using training signal from raw data, (ii) federated transfer learning (Fed-TL) using training signal from model-interpretable representation, and (iii) federated reinforcement learning (Fed-RL) using training signal from preference. As next, we further illustrate the three types of FL approaches in lens of the introduced general FL framework in Section 2.2 (shown in Table 4).

-(i) Federated supervised learning using training signal from raw data. It is currently the most widely used FL paradigm. Firstly, for the $\mathcal{L}$**.update**, each FL clients update its local model using optimization methods such as SGD [124] or Adam [125] on its private dataset. Then, the FL clients share their gradients or model parameters to the Fed-ST server, before which the FL client can apply the privacy mechanism in the $\mathcal{L}$**.prepare**. After that, the FL server performs weighted aggregation (i.e., $\mathcal{G}$**.aggregate**), where the weights for FL clients are typically based on the size of their datasets. Finally, the FL server broadcasts the updated global model back to all participating FL clients.

-(ii) Federated transfer learning using training signal from representation: It enables FL clients to use diverse model-interpretable training signals (e.g., logits, embeddings and proxy model) during the local update $\mathcal{L}$**.update**. For example, in federated knowledge distillation [126], the local update process can incorporate the discrepancy between the local model and the global model as an additional supervisory signal. The Fed-TL also allows a more flexible aggregation $\mathcal{G}$**.aggregation** on the FL server. As the example in Fig. 5(b), when the FL clients upload the prompt embeddings in high-dimensional vector form, the FL-TL server can utilize an attention mechanism [127] to assign weights to each embedding and then compute their clustering center to obtain aggregated results.

-Federated reinforcement learning using training signal from preference: It provides a way to align the heterogeneous preferences from FL participants. In $\mathcal{G}$**.aggregation**, the Fed-RL can either aggregate the decision policies (e.g., Q-tables) or just follow the parameter aggregation approaches as Fed-ST. The Fed-RL facilitates using high-level, abstract preference signals to improve reasoning capability for large models. For instance, reinforcement learning from human feedback (RLHF) [128] can train the LLMs through user preferences (users select one of two LLM outputs as the preference). In the LLM-as-Judger system [129], Fed-RL can also take the

Table 4 Comparison of the three federated training paradigms through the FL framework in Section 2.2

	$\mathcal{L}$.update	$\mathcal{L}$.prepare	$\mathcal{G}$.aggregation
Fed-ST [19,121]	SGD, Adam, etc.	Gradients, Random Seeds, etc.	Weighted Average
Fed-TF [46,122]	KD, CLIP, etc.	Logits, Models, etc.	Weighted Average
Fed-RL [76,123]	Q-learning, DPO, etc.	Q-table, RL models, etc.	Weighted Average

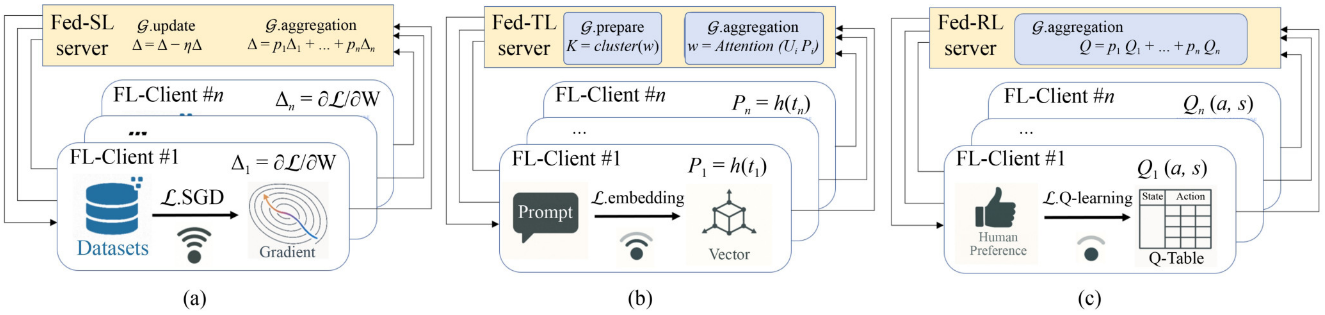


Fig. 5 Examples for training signals from raw data, learned representation and preference. (a) Signal from Raw Data (federated supervised learning); (b) signal from Representation (federated transfer learning); (c) signal from preference (federated reinforcement learning)

assigned credits of tokens from AI models to improve reasoning capabilities of rLLMs.

To summarize, we introduce a novel taxonomy for federated training of reasoning-oriented LLMs, grounded in the nature of training signal formats. The proposed taxonomy encompasses three typical FL paradigms, i.e., federated supervised learning based on training signals from raw data, federated transfer learning which leverages learned representations, and federated reinforcement learning driven by preference-based signals. In addition, we facilitate a comparative understanding of these approaches through the general FL framework introduced earlier in the survey. In the following sections, we examine the representative studies corresponding to each training signal in more detail.

4 Training signal from raw data

In this subsection, we introduce representative FL approaches that take training signals derived from raw datasets and usually share gradients during training, i.e., federated supervised learning (Fed-SL).

The Fed-SL can mainly enhance reasoning capabilities of rLLMs from two key perspectives: (i) Domain-specific reasoning. In domain-specific scenarios, such as industrial code generation, access to high-quality private datasets is essential for enabling the effective domain-specific reasoning [130]. However, such datasets cannot be shared directly due to privacy constraints. Fed-SL offers a practical solution by allowing rLLMs to learn from distributed, high-quality, private data while preserving data privacy. (ii) Instruction following. A presupposition for reasoning is to understand the user's instructions. However, user instructions usually vary significantly from application to application [131]. Fed-SL can facilitate adaptation to such heterogeneity by enabling rLLMs to better exhibit instruction following behaviors in a federated manner.

In the following, we review two representative categories of Fed-SL techniques in federated supervised pretraining and federated supervised instruction tuning from three aspects, i.e., model capability, communication cost and privacy protection. It is worth noting that the pre-training can also serve as the full-parameter fine-tuning from the technique perspective and we do not explicitly distinguish these approaches within the context of Fed-SL based on training signal from raw data.

4.1 Federated supervised pre-training

This category of approaches primarily aims to pre-train LLMs using Fed-SL approaches or to fine-tune them in a pretraining-style manner. The challenges in this setting align with those encountered in previous Fed-SL research for smaller language models, including: (i) data heterogeneity, (ii) privacy preservation, and (iii) communication overhead. In particular, the massive scale of LLMs —often reaching billions of parameters —makes transmitting full model updates prohibitively expensive in terms of communication overhead. Consequently, existing studies on the federated supervised pre-training of rLLMs primarily focus on mitigating significant communication bottlenecks through novel federated algorithmic designs (shown in Table 5).

From federated SGD to federated ZOO. **Firstly**, one line of work in Fed-ST approaches build upon conventional FL algorithms by either directly adopting existing methods [24] or proposing new model aggregation techniques [25]. A key feature of these methods is that they continue to rely on optimization techniques such as stochastic gradient descent (SGD) [124], which implies that such methods have to transmit gradients or parameters at a scale comparable to the full model size, leading to prohibitive communication costs. Consequently, they face significant scalability bottlenecks, particularly when applied to large-scale models or federated learning scenarios involving numerous clients. In [25], Yang et al. investigate the cross-cloud federated training (CCFT) of LLMs using an SGD-based optimization framework. The authors proposed an asynchronous gradient update approach to optimize the communication cosh, which takes the FL client's loss value as the weight for aggregation. The asynchronous updates can be formally described as,

Table 5 Summary of the representative federated supervised pre-training approaches

	FL-Setup	#Client	$\mathcal{G}$.aggergation	$\mathcal{G}$.update	$\mathcal{L}$.update	$\mathcal{L}$.prepare	Basic Model	Comm. Tech.	Privacy Tech.
LLaVAFL [24]	*Cross-Devices*	20	*FedAvg*	*SGD*	*SGD*	–	Resnet-50	–	–
CCFT [25]	*Cross-Silos*	3	*Softmax Loss*	*SGD*	*SGD*	–	–	*Async. Update*	*DP*
Photon [26]	*Cross-Silos*	16	*FedAvg*	*SGD*	*SGD*	*Grad-Clipping*	7B Model	*RDMA*	*DP*
FedRDMA [29]	*Cross-Silos*	16	*FedAvg*	*SGD*	*SGD*	–	GPT-2	*RDMA*	–
Ferret [27]	*Cross-Devices*	200	*FedAvg*	*FOO*	*SGD*	*Projection*	LLaMA-3B	*Random Seeds*	–
FedCyBGD [31]	*Cross-Devices*	64	*FedAvg*	*CyBGD*	*CyBGD*	*Block-Pruning*	LLaMA-7B	–	–
FedMeZo [32]	*Cross-Silos*	8	*FedAvg*	*ZOO*	–	*ZOO*	LLaMA-7B	*Random Seeds*	–
FedKSeed [33]	*Cross-Devices*	738	*FedAvg*	*ZOO*	*ZOO*	*Random Seeds*	LLaMA-3B	*Random Seeds*	–
FedFeedSign [28]	*Cross-Devices*	25	*FedAvg*	*ZOO*	*ZOO*	*Random Seeds*	RoBERTa	*Random Seeds*	–
FwdLLM [30]	*Cross-Devices*	500	*FedAvg*	*ZOO*	*ZOO*	*Random Seeds*	LLaMA-7B	*Random Seeds*	–

$$\Theta_G^{(t+1)} = \Theta_G^{(t)} + p_c(\Theta_c^{(t)} - \Theta_G^{(t)}),\ p_c = \frac{\mathrm{e}^{-l_c}}{\sum_{j\in C}\mathrm{e}^{-l_j}}, \tag{5}$$

where l_c is the loss value from local FL clients and $\Theta_G^{(t+1)}$ and $\Theta_G^{(t)}$ are the model parameters in the $(t+1)_{\text{th}}$ and t_{th} training round, respectively. They conduct experiments on three major cloud platforms, including AWS, Google Cloud, and Azure.

Secondly, to address the communication bottleneck of previous SGD-based FL-ST approaches that require the aggregation of all model parameters, Wang et al. [31] proposed a communication-efficient approach that selectively updates a subset of parameters. They replace the standard SGD with a Cyclic Block Gradient Descent (CyBGD) strategy, which can reduce communication overhead while maintaining competitive model performance. Specifically, FedCyBGD assigns each FL client a distinct block of model parameters, and the FL clients update in a predefined cyclic order. Then, updated parameter blocks are aggregated to refine the global model. The authors further reduce the communication cost by a hybrid parameter compression approach $C(\cdot)$ for LLMs, which is formally as,

$$C(\Theta_c^{(t)}) = \{\mathbb{P}_1(\Theta_1^{(t)}), \ldots, (\Theta_{c-1}), \Theta_c, \ldots, \mathbb{I}_n\Theta_n\}, \tag{6}$$

where $\mathbb{P}$ is a standard compression method, and $\mathbb{I}$ is an indicator variable for the randomly dropping.

Recently, to further reduce the communication cost, researchers [33] have utilized the zero-order optimization (ZOO) strategies, which takes the function values to approximate the gradients as follows,

$$\nabla\mathcal{L}(x) \approx \frac{1}{m}\sum_{i=1}^{m}\frac{\mathcal{L}(x+a\mathbf{z}) - \mathcal{L}(x-a\mathbf{z})}{2\|a\|}a\mathbf{z}, \tag{7}$$

where $\mathcal{L}(\cdot)$ is the loss function and $\mathbf{z}\in\mathbb{R}^d$ is the random perturbation from a normal *Gaussian* distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. As the $\mathbf{z}$ is indeed the basis for parameter updates. In ZOO strategies, as the gradient estimation is performed based on a set of biases random sampled from the *Gaussian* distribution, we can leverage this property in FL and only transmit scalar values for aggregation. Specifically, the FL server first generates a set of random seeds $S = \{s_1, \ldots, s_k\}$, which are shared with all FL clients. Then, each FL client uses these seeds to deterministically sample a set of *Gaussian* basis vectors $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$. By ZOO, each FL client can perform a local update and estimate the corresponding gradient coefficients $A_c = \{a_{c,1}, \ldots, a_{c,k}\}$ based on basis vector $\mathbf{Z}$. Finally, the FL server aggregates the scalar coefficients, i.e., $a_{G_i} = \sum_{c\in C} a_{c,i}, (i = 1, \ldots, k$ from all clients to obtain a global gradient estimation. As only scalar values, instead of full gradient parameters, are transmitted during training, the communication cost in Fed-ST for `rLLMs` can be significantly reduced. Addressing communication overhead, the ZOO-like FL has emerged as a promising paradigm for federated supervised learning within the context of `rLLMs` [28,30,32,33]. We also compare these techniques in terms of communication efficiency in Fig. 6.

Next, we briefly review and discuss the privacy risks and corresponding countermeasures for *federated supervised pre-training*. On the one hand, federated pre-train approaches [24–26] are subject to privacy risks similar to those in conventional FL, including gradient leakage attacks [132] and backdoor attacks [133]. As limited research have been done on privacy risks for full-parameter federated training for `rLLMs`, we provide a preliminary analysis of potential vulnerabilities in this context. Notably, experimental results in [134] suggest that when the FL models are initialized with pre-trained parameters, the risk of gradient leakage from gradients can be significantly reduced, indicating that `rLLMs` may be inherently more robust to gradient-based privacy attacks than smaller `LMs` during supervised fine-tuning. On the other hand, federated training approaches based on Zero-Order Optimization (ZOO) only transmit a limited number of scalar values (i.e., random seeds) during the FL training process [32,33], which contrasts with the previous SGD-based methods. In this case, whether it is possible to reconstruct an FL client's private data through transmitted random seeds (similar to gradients) remains an open research question.

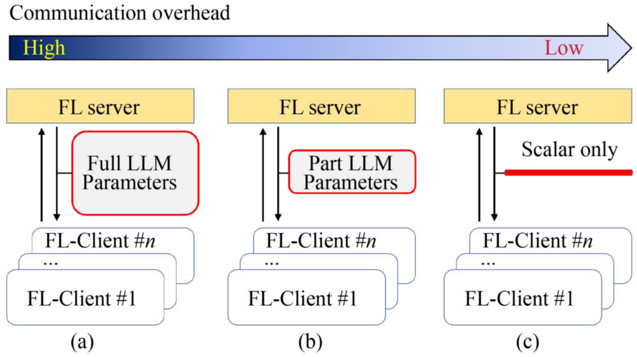


Fig. 6 Federated Supervised Learning: From SGD to ZOO. (a) Fed-SGD; (b) Fed-CyBGD; (c) Fed-ZOO

4.2 Federated supervised instruction-tuning

Federated instruction tuning (FIT) enhances `rLLMs`' ability to handle diverse user instruction tasks based on Fed-ST. The objective of FIT is to enable the model $\mathcal{M}_\Theta$ to reason and then output the answer tokens Y, given an instruction $\mathcal{I}$ and a contextual input X, which can be formally described as,

$$\Theta^* = \arg\min_{\Theta}\mathbb{E}_{(\mathcal{I},X,Y)\sim\mathcal{D}_{\text{Instr}}}[\mathcal{L}(\mathcal{M}_\Theta(\mathcal{I},X),Y)], \tag{8}$$

where $\mathcal{I}$ is the user's task instruction, X is the input context and Y denotes the output answer of `rLLMs` $\mathcal{M}_\Theta$. Enabling `rLLMs` to generalize to unseen instructions is fundamental to developing robust reasoning capabilities for diverse real-world tasks. However, instruction data provided by `rLLM` users is often fragmented across numerous end-user devices and exhibits substantial heterogeneity in quality and intent across different application scenarios [34]. In this context, it is crucial to utilize the federated instruction tuning to learn as more as the private instruction triples $(\mathcal{I}, X, Y)$ across numerous `rLLM` users. *Therefore, existing studies in this area primarily focus on improving model capability.*

From data heterogeneity to data-diversity-aware. **Firstly**, in the seminal work [34], Zhang et al. extended the Fed-ST approach to instruction tuning by leveraging distributed user instructions, thereby avoiding the expensive costs of instruction collection. They also revealed the typical data heterogeneity in user instructions. For example, QA tasks usually require concise and fact-based responses, whereas writing instructions prioritize coherent and logical text. Though data heterogeneity is typically regarded as the key obstacle to effective FIT [34], Wang et al. [135] have experimentally demonstrated that, in the context of FIT, the data heterogeneity does not showcase the monotonic relationship to instruction following ability. Notably, they have also provided an interesting insight that the data diversity can significantly influence the `rLLM`'s instruction following capabilities, i.e., *it is the data diversity matters in federated instruction tuning*.

Then, recent research [135] has increasingly emphasized data-centric approaches to federated instruction tuning, highlighting the importance of data diversity and quality across user instruction domains. These approaches typically involve actively selecting a core subset of available instructions as an enhancement. We categorize data-centric Fed-IT approaches into three groups: (i) Heuristic Scoring-based Data Selection [35], (ii) Clustering-based Data Selection [36,135], and (iii) LLM-Generation-based Data Selection. In the following, we review the representative studies from each category.

(i) Heuristic-scoring-based: It first employs a scoring function to assess the quality or diversity of the instruction data and then selects a subset of instructions for FL training based on the score ranking. For example, authors in [35] adopt *Instruction-Response Alignment* (IRA) as the score function, formally described as,

$$\mathrm{IRA}((X,\mathcal{I}),\Theta)=\mathcal{L}(X;\Theta)-\mathcal{L}((X,\mathcal{I});\Theta), \tag{9}$$

where $\mathcal{L}(X;\Theta)$ and $\mathcal{L}((X,\mathcal{I}))$ are the loss function with and without instructions, respectively.

(ii) Clustering-based: It aims to maximum the domain coverage (i.e., data diversity) across FL clients by solving a clustering optimization problem. For example, authors in [135] formulate an optimization problem to maximum the data diversity and minimize the communication overhead at the same time, which is formalized as,

$$\arg\min_{\mathcal{P}}\left\{\sum_{c\in C}|\mathcal{P}_c|-\frac{1}{|D^d|}\sum_{d\in D^d}\max_{p\in\mathcal{P}}\mathrm{sim}(d,p)\right\}, \tag{10}$$

where $\mathcal{P}$ are the cluster centers and $\mathrm{sim}(\cdot,\cdot)$ is the function (e.g., cosine function) measuring similarity between the in-domain data point d from the domain dataset D^d and the centroid p.

(iii) LLM-Generation-based: It uses `LLMs` or `rLLMs` to generate synthetic instruction data to make further fine-tuning, which extends the diversity of instruction data as well. For example, Zhan et al. [38] propose FewFedPIT, which utilizes real instructions together with synthetic instructions to update the local model as,

$$\Theta_c=\beta\cdot\Theta_c^l+(1-\beta)\cdot\Theta_c^g, \tag{11}$$

where Θ_c^l and Θ_c^g are the model parameters trained on local private instruction data and generated synthetic instructions of client c, respectively.

Finally, we make a discussion on the data-diversity aware FL with previous FL approaches from the perspective of client or data selection. As shown in Fig. 7, in vanilla FL [18,19], the server typically selects a random subset of clients to perform training on their local data. However, this approach overlooks the inherent data heterogeneity across FL clients. As a remedy, prior studies have explored client selection aware FL (C-Select FL) [136,137], which evaluates each FL client's contribution [138,139] and prioritizes those with higher contributions dataset in the training process. Furthermore, in the federated instruction tuning, the unique characteristics of instruction data and user tasks necessitate an even stronger emphasis on the data diversity [34,36,135]. This leads to a new kind of FL approaches that performs data-level selection at a finer granularity before the training process. It considers which specific data points within FL clients are most beneficial for FL training. We envision that this fine-grained data selection based FL approaches should be promising for a broader applications beyond the Fed-IT, particularly when both the data diversity and FL tasks generalization play the important roles.

Next, we discuss the privacy risk in federated instruction tuning. As stated in the AI alignment paradox [140], "*More virtuous AI is more easily made vicious*", it highlights a fundamental safety issue, while instruction tuning improves the `LLM`'s ability to understand and follow user instructions, it simultaneously increases the risk that malicious attacker could exploit `rLLMs` to follow harmful instructions. Representatively, Ye et al. [141] propose an FL server-side defense approach for *Fed-IT*, in which the FL server performs additional fine-tuning using a defense dataset after each aggregation step to mitigate the influence of potentially poisoned model updates from malicious clients. As discussed above, most existing Fed-IT approaches have primarily focused on data-diversity-aware federated training. However, identifying and filtering the malicious instructions or outputs contributed by compromised FL clients remains an open and critical research challenge.

4.3 Summary of signal from raw data

We summarize FL techniques based on training signals from raw data from aforementioned three key aspects: model capability, communication efficiency, and privacy preservation. Firstly, concerning model performance, the focus of the federated learning

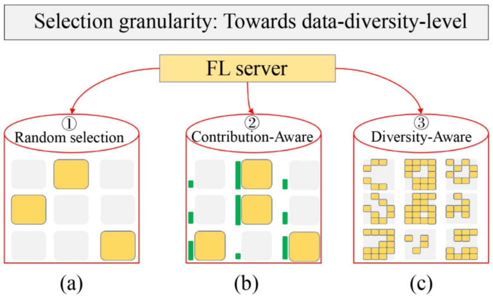


Fig. 7 Towards Data-Diversity-Aware Fed-ST. (a) Vallina FL; (b) C-Select FL; (c) D-Select FL

community has shifted from client-centric selection to data-diversity-aware approaches, which aim to enhance instruction-following capabilities across diverse downstream tasks by selecting diverse subsets of instruction data prior to training. Secondly, to address the prohibitive communication overhead associated with full-parameter transmission during federated pretraining, the ZOO-based training approaches [32,89] have been widely adopted, which enable the FL training by transmitting only random seeds and scalar coefficients, thereby significantly reducing communication costs. Meanwhile, how to ensure model accuracy in federated supervised pre-training using random seeds remains a key research challenge. Finally, in the privacy preservation context, both federated supervised pre-training and federated instruction tuning introduce new challenges. For federated pre-training, an open research question is whether using random seeds could inadvertently leak raw training data. In the case of federated instruction tuning, a critical concern is whether we can prevent the `rLLMs` from learning to follow harmful instructions from malicious clients.

5 Training signal from representation

In this subsection, we introduce federated transfer learning (Fed-TL) approaches of `rLLMs`, which improves reasoning capabilities based on the `rLLMs` interpretable training signals, e.g., learned representation, adapter modules, logits or proxy models.

We divide the Fed-TL for `rLLMs` into three categories according to the format of training signal during FL training process, including: (i) Federated Prompt Learning using shared representations (i.e., embeddings), (ii) Federated Adapter Learning using collaboratively trained adapter modules, and (iii) Federated Knowledge Distillation using the transferred knowledge (e.g., logits or proxy small models). Next, we briefly discuss how these three Fed-TL techniques enhance the reasoning capabilities of `rLLMs`. Firstly, Federated Prompt Learning reformulates user discrete or continuous prompts into formats that are more interpretable and actionable for `rLLMs`, thereby incentivizing a more effective reasoning process [142] just like the federated instruction tuning [34]. Secondly, Federated Adapter Learning allows `rLLMs` to incorporate the domain-specific knowledge in a plug-and-play fashion, therefore improving the abilities for the domain-specific reasoning as well. Lastly, since smaller language models are usually highly optimized for the specific tasks, Federated Knowledge Distillation enables the `rLLMs` to align with the task-specific strengths of small language models by distilling their knowledge through the proxy models or logits, which provide a privacy-preserving but communication-efficient federated training manner. We review representative studies and discuss emerging research trends within each topic area, with a particular focus on three key aspects: model capability, communication cost, and privacy protection.

5.1 Federated prompt learning

Prompt learning is designed to reformat input queries into certain templates, like `[vector]` + `Reviews[The movie was great]` + `Sentiment[Positive]`, which elicit the `rLLM's` desired behaviors (e.g., reasoning), without changing the model's internal parameters [7]. Similarly, Federated Prompt Learning (Fed-Prompt) [40,41,45] aims to jointly learn a prompt function that transforms heterogeneous user inputs into formats more suitable for `rLLMs` across FL clients, facilitating a consistent and effective reasoning process for diverse user queries, i.e., existing research has primarily focused on improving model performance in this direction (in Table 6).

From discrete prompts to continuous prompts. **Firstly**, as `rLLMs` possess the in-context learning capabilities [4,81], they can learn directly from the prompt templates provided as examples. In this case, a straightforward approach to achieve *Fed-Prompt* is sharing and aggregating the prompt data from FL clients in the training process [40,41]. In [41], the authors proposed a knowledge compendium based approach, called FICAL, where each FL client generates prompts based on its local knowledge compendium and then the FL server concatenate them as the global compendiums.

Table 6 Summary of the representative federated prompt learning approaches

	FL-Setup	#Client	Prompts	$\mathcal{L}$.update	$\mathcal{G}$.aggregate	$\mathcal{G}$.update	Basic Model	Privay Tech.
Fed-SP-SC [39]	*Cross-Devices*	–	*Discrete*	–	*Direct*	–	GPT-3.5	–
FedFSCD [40]	*Cross-Silos*	–	*Discrete*	–	*Direct*	–	GPT-4	*Prompt Inject.*
FICAL [41]	*Cross-Devices*	8	*Discrete*	–	*Direct*	*Embedding*	LLaMA3-8B	*Compendium*
PromptFolio [42]	*Cross-Devices*	100	*Continuous*	*Portf. Opt.*	*Direct*	CoOp-like	CLIP	–
FedAPT [43]	*Cross-Devices*	30	*Continuous*	*SGD*	*Direct*	*SGD*	CLIP	–
FedPepTAO [44]	*Cross-Devices*	100	*Continuous*	*Adam*	*Direct*	*Mom*	LLaMA-7B	–
PromptFL [45]	*Cross-Devices*	64	*Continuous*	*SGD*	*Direct*	*SGD*	CLIP	–
FedDTPT [46]	*Cross-Devices*	10	*Continuous*	–	*Clustering*	–	DeepSeek-V2	–
PLAN [47]	*Cross-Silos*	6	*Continuous*	*SGD*	*Model-Based*	*SGD*	CLIP	–
FedBPT [48]	*Cross-Devices*	20	*Continuous*	*Black-box Tuning*	*Model-Based*	*CMA-ES*	LlaMA2-7B	–
PFPT [143]	*Cross-Devices*	80	*Continuous*	*Adam*	*Model-Based*	*Bi-Opt.*	ViT	–

After that, the global compendiums are encoded into embeddings and stored in a vector database for future use. Similarly, Seo et al. [41] proposed a security-aware framework where each FL client leverages `LLMs` to transform its local prompts into a standard format and applies prompt injection defense mechanisms against malicious attack. Then, the processed prompts are aggregated on the FL server.

Recently, research efforts have increasingly focused on *continuous prompts* in embedding vector format (*a.k.a. soft prompts*) rather than the discrete prompts in raw-text format. Continuous prompts are dense, trainable representations and are generally considered to provide a stronger privacy preservation level as they are less human-interpretable. From the perspective of federated training, continuous prompts enable more flexible aggregation strategies $\mathcal{G}$.**aggregate**, which mainly includes: (i) *Direct aggregation*, (ii) *Clustering-based Aggregation*, and (iii) *Model-based Aggregation*. We review the representative studies for each as follows.

(i) Direct aggregation: Given that word embeddings are often assumed to exhibit additive properties [144], previous studies [42–44] have adopted a simple yet effective aggregation strategy, i.e., taking a FedAvg-like [121] method to average prompt embeddings across FL clients $\{\mathbf{P}_1,\ldots,\mathbf{P}_n\}$, which can be formalized as,

$$\mathbf{P}_G=\sum_{c\in C}\frac{D_c}{\sum_{j\in C}|D_j|}\mathbf{P}_c, \tag{12}$$

where $\mathbf{P}_G$ is the global aggregated prompt vectors and $|D_c|$ denotes the total number of data samples of FL client c. In [42], the authors theoretically and empirically demonstrate that combining multiple FL clients leads to better performance than using prompts from single client.

(ii) Clustering-based aggregation: Since prompt embeddings are inherently high-dimensional vectors, we can also use clustering-based methods to compute the centroid of FL clients' prompt embeddings as the aggregated result. For example, Wu et al. [145] propose FedDTPT, a clustering based approach to aggregate prompt embeddings. Specifically, FedDTPT first computes the semantic similarity among prompt vectors and then employs the classical DBSCAN as clustering method for semantic aggregation.

(iii) Model-based aggregation: Beyond direct average or clustering, model-based prompt aggregation is also widely adopted in previous work [47,48], where neural networks or probabilistic models are adopted to perform a more expressive fusion of federated prompt representations. For instance, Gong et al. [47] proposed a federated prompt learning framework that employs attention-based neural networks for prompt aggregation, where the attention-based model is jointly trained using the FedAvg algorithm simultaneously. Sun et al. [48] proposed a model-based prompt learning framework called FedBPT, which allows the FL server to execute the Covariance matrix adaptation evolution strategy (CMA-ES) [146], a gradient-free optimization algorithm, to aggregate local prompt distributions from FL clients without relying on explicit gradients.

In the following, we briefly discuss the privacy protection in federated prompt learning, particularly in scenarios involving discrete prompts, where FL clients directly upload textual prompts to the FL server [39,41]. This setting introduces new privacy and safety challenges, such as prompt injection attacks [40], in which adversarial clients craft malicious prompts that induce the model to exhibit unintended or harmful behaviours. In response, Seo et al. [40] propose a direct defense framework in which `rLLMs` assess and filter prompts based on their internal reasoning capabilities. Besides, the authors in [141] present a complementary defense strategy, where the FL server performs further training on a curated defense dataset after aggregating prompts from potentially malicious clients, helping to mitigate the adverse effects of injected prompts.

5.2 Federated adapter learning

Adapter Learning (or Adapter Tuning) is a parameter efficient fine-tuning approach that takes the `LLMs` as frozen black- boxes and trains only limited additional adapter parameters for various downstream tasks. For *Federated Adapter Learning* (Fed-AL), it trains the adapters across FL clients in a distributed manner, thereby enhancing the task-specific reasoning capabilities of `rLLMs`. The benefits of Fed-AL are twofold. Firstly, it updates only adapter parameters, is resource-efficient, and significantly reduces computational, communication, and memory overhead. Secondly, it can serve as plug-and-play modules for diverse downstream tasks, making it highly flexible and easily incorporable within the `rLLMs` architecture. As shown in Fig. 8, prior Fed-AL approaches can be divided into three categories based on the placement of adapter modules within the Transformer architecture of `rLLMs`, including (i) Layer-Internal, (ii) Layer-External, and (iii) Hybrid Adaptation. In the following, we review representative studies in each category (summarized in Table 7), with a primary focus on improving model capability.

From internal adapters to external adapters. **Firstly,** as illustrated in Fig. 8(a), the federated internal adapter approaches primarily utilize the Low-Rank Adaptation (LoRA) framework [34,36,56,61,149], where low-rank matrices are injected into `rLLMs` to enable parameter-efficient fine-tuning. The main idea of LoRA is that the weight matrices in `rLLMs` are intrinsically low-rank and can be replaced by the product of two smaller matrices, Θ_A and Θ_B, which can be formally described as follows,

$$\Theta=\Theta+\Delta\Theta,\quad \Delta\Theta=\Theta_A\cdot\Theta_B, \tag{13}$$

where $\Theta\in\mathbb{R}^{p\times q}$ is the frozen parameters, and the trainable parameters $\Theta_A\in\mathbb{R}^{q\times r}$ $\Theta_B\in\mathbb{R}^{r\times p}$ $(r\ll p,q)$. In Fed-AL, FL clients can share only the adapter rather than the full model parameters, enabling communication-efficient joint training. However, how to aggregate these adapters from FL clients on the FL server is non-trivial [56] and we review representative studies from the perspective of adapter aggregation strategies in the following. In [148], the authors propose a LoRA based federated parameter-efficient tuning approach and set trainable parameters less than 1%. They directly treat trainable parameters $\Theta^{\mathrm{LoRA}}=\{\Theta_A,\Theta_B\}$ as previous in FL and aggregate them using FedAvg [121], i.e.,

$$\Theta^{\mathrm{LoRA}}=\sum_{c\in C}|D_c|/D_C|\cdot\Theta_c^{\mathrm{LoRA}}. \tag{14}$$

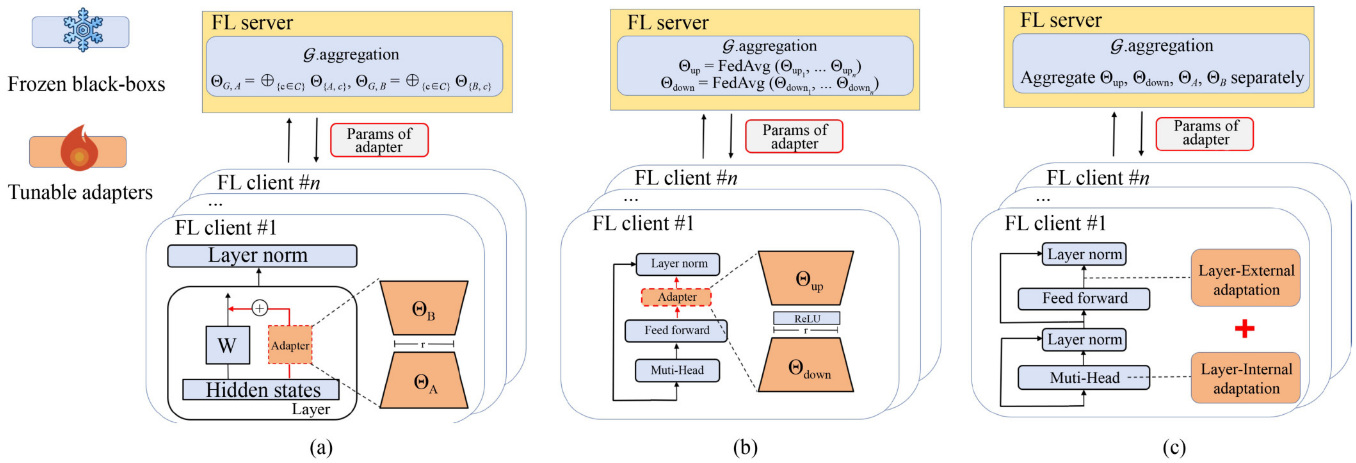


Fig. 8 Federated adapter tuning: from internal adaption to external adaption. (a) Layer-Internal adaptation (b) layer-external adaptation; (c) hybrid adaptation

Table 7 Summary of the representative federated adapter learning approaches

	FL-Setup	#Client	Type	Position	Architect.	Basic Model	Privacy Tech.
Fed-IT [34]	*Cross-Devices*	100	*Internal*	*Inside ATT, FFN*	*LoRA*	LLaMA-7B	-
FFA-LoRA [61]	*Cross-Silos*	3	*Internal*	*Inside ATT, FFN*	*LoRA*	RoBERTa-Large	*DP*
FLoRA [56]	*Cross-Devices*	10	*Internal*	*Inside ATT*	*LoRA*	LLaMA-7B	*DP, HE*
FedHDS [36]	*Cross-Devices*	200	*Internal*	*Inside ATT, FFN*	*LoRA*	LLaMA-3B	*DP*
FibecFed [63]	*Cross-Devices*	10	*Internal*	*Inside ATT, FFN*	*LoRA*	LLaMA-7B	–
FedIT-U2S [37]	*Cross-Silos*	5	*Internal*	*Inside ATT, FFN*	*LoRA*	Vicuna-7B	–
Fed-SB [58]	*Cross-Devices*	25	*Internal*	*Inside ATT, FFN*	*LoRA*	LLaMA3-3B	*DP*
FedDPA [62]	*Cross-Silos*	8	*Internal*	*Inside ATT*	*LoRA*	LLaMA-7B	–
FedMT [57]	*Cross-Silos*	2	*Internal*	*Inside ATT*	*LoRA*	LLaMA3-8B	–
FedPA [147]	–	–	*Internal*	*Inside MLP*	*LoRA*	–	*DP*
FedDB [59]	*Cross-Silos*	50	*Internal*	*Inside MLP*	*LoRA*	ViT	*HE, SMC*
FedPEFT-A [148]	*Cross-Silos*	8	*External*	*After FFN*	*Pfeiffer*	ViT-B	*DP*
FedDAT [72]	*Cross-Silos*	5	*External*	*After FFN*	*Houlsby*	ViLT	–
FedPIA [60]	*Cross-Devices*	200	*External*	*After ATT, FFN*	*Houlsby*	ViLT	–
FedMCP [64]	*Cross-Silos*	6	*External*	*After Transformer*	*Houlsby*	RoBERTa	–
FedFMSL [65]	*Cross-Devices*	15	*Hybrid*	*After EC, TF*	*LoRA,Gate*	CLIP	–

However, such direct aggregation is not reasonable for the LoRA based federated adapter learning, as

$$\sum_{c\in C}\Theta_{B,c}\cdot\Theta_{A,c} \neq \sum_{c\in C}\Theta_{B,c}\cdot\sum_{c\in C}\Theta_{A,c}. \tag{15}$$

Therefore, more FL aggregation solutions are proposed to tackle above challenges. As an initial try, Sun et al. [61] propose the FFA-LoRA, an approach tailored for federated aggregation. Specifically, FFA-LoRA initializes the low-rank parameter Θ_A from Gaussian and frozen in the FL server aggregation phase $\mathcal{G}$.**aggregate**. Thus, only the other parameters Θ_B need to be aggregated in server, i.e.,

$$\Theta_G = \Theta + \Delta\Theta_G, \quad \Delta\Theta_G = \Theta_{A,0}\cdot\sum_{c\in C}\Theta_{B,c}, \tag{16}$$

where $\Theta_{A,0}$ denotes the fixed parameters in LoRA. In [56], the authors proposed an alternative aggregation approach for LoRA-based federated adapter learning, termed FLoRA, which further utilize both Θ_A and Θ_B in the FL aggregation phase. Specifically, the $\Theta_{A,c}$ and $\Theta_{B,c}$ matrices from each FL client c are firstly concatenated on the FL server and then multiplied to form the aggregated update, ensuring an equivalent federated extension of the original LoRA formulation, which is formalized as,

$$\sum_{c\in C}\Theta_{B,c}\cdot\Theta_{A,c}=(\oplus_{c\in C}\Theta_{B,c})\cdot(\oplus_{c\in C}\Theta_{A,c}), \tag{17}$$

where the symbol ⊕ denote the concatenate operation. Therefore, the FLoRA preserves the low-rank structure while enabling cross-client training.

Recently, some studies [36,60,64,72] have shifted from LoRA-based adapters to Houlsby-based external adapters [118] (as in Fig. 8(b)) in the context of federated adapter learning for `rLLMs`. This transition is primarily motivated by the fact that Houlsby-based adapters do not require breaking the inside structure of `rLLMs`, making it easier to integrate. As plug-and-play modules, external adapters can be easily extended to multiple downstream reasoning tasks. In contrast, LoRA-based adapters often struggle to achieve such flexibility due to their tighter coupling with internal model parameters [118]. The main idea of Houlsby-based adapter is to inject an adapter with two part of trainable parameters $\Theta_{up},\Theta_{down}$ after the FFN of each Transformer block, and then the activations can be calculated as,

$$h'=h+\sigma(h\cdot\Theta_{down})\Theta_{up}, \tag{18}$$

where h is the normalized output of the FFN and σ is the activation function. Representatively, Zhao et al. proposed *FedMCP* [64], a Houlsby-style federated adapter learning framework designed with a dual-adapter architecture. In FedMCP, each FL client holds the private adapters $\Theta_{up}^{pri},\Theta_{down}^{pri}$ and the global adapters $\Theta_{up}^{glob},\Theta_{down}^{glob}$, where global adapters are uploaded to the FL server and aggregated via FedAvg [121], while the private adapters remain local. Both adapters are jointly utilized during inference, allowing the model to benefit from shared global knowledge while preserving personalized reasoning capabilities, which is formally as follows,

$$h'=h+\frac{1}{2}\sigma(h\Theta_{down}^{pri})\Theta_{up}^{pri}+\frac{1}{2}\sigma(h\Theta_{down}^{glob})\Theta_{up}^{glob}, \tag{19}$$

where $\Theta_{up}^{pri},\Theta_{down}^{pri}$ and $\Theta_{up}^{glob},\Theta_{down}^{glob}$ are the local private adapter and global public adapter, respectively. Instead of directly aggregating the trainable parameters of adapters, Saha et al. [60] proposed a permutation matrix-based aggregation approach to align adapter parameters between the server and clients.

5.3 Federated knowledge distillation

Knowledge distillation (KD) [150] provides a flexible mechanism for transferring knowledge between machine learning models. Federated Knowledge Distillation (Fed-KD) extends this concept to privacy-sensitive settings by allowing each client to learn from its local private dataset and transmit distilled knowledge, rather than raw data or full model parameters, to improve the reasoning capability of large models collaboratively. Next, we review representative studies on Fed-KD for large models (in Table 8), which adopt various forms of knowledge, including *logits*, *soft labels*, and *proxy models* and so on.

Knowledge formats: from activations to models. (i) Activations-based: To address the model size mismatch between the FL server and clients, Dong et al. [66] utilize hidden states (i.e., activations) in neural networks as the knowledge to update the global model. (ii) Logits based: In [67], Fan et al. proposed a federated knowledge transfer framework for small and large models, called FedMKT, which leverages *logits* (values before the softmax function used for predicted probabilities) to enable bidirectional knowledge transfer between a large server-side model and client-side lightweight models, largely reducing the communication cost. (iii) Predictions based: In [71], the authors assume that the FL server and clients share a same public dataset and then the server and client can align their models through predictions (i.e., soft labels) in the federated knowledge distillation. (iv) Adapter based: Fan et al. [68] also proposed a parameter-efficient federated knowledge distillation framework that fine-tunes both small and large models, where knowledge transfer is performed through adapter parameters. (v) Proxy Model based: Besides, the proxy models, which are compressed from a large model, can also be used to update large language models on the FL server [73]. In summary, in Fed-KD of large models, existing studies have explored various knowledge formats as alternatives to traditional gradient sharing for global and local model updates. However, which formats provide better trade-off between accuracy, efficiency, and privacy remains an open question.

We make a discussion on privacy risk in federated knowledge distillation (Fed-KD) below. As aforementioned, a primary feature of Fed-KD for `rLLMs` is that existing studies [47,66–68,73] can utilize

Table 8 Summary of the representative federated knowledge distillation approaches

	FL-Setup	#Client	Knowl. Format	Teacher	Student	Mutual	$\mathcal{G}$.distil	$\mathcal{L}$.distil	Loss Func.	Basic Model
FEDSP [66]	*Cross-Devices*	10	*Activations*	*Large*	*Small*	–	√	–	$\mathcal{L}_2$	GPT-2-1.5B
FedMKT [67]	*Cross-Devices*	4	*Logits*	*Large*	*Small*	√	√	√	$\mathcal{L}_{CE}$, *Loss*	LLaMA2-7B
FedID [71]	*Cross-Devices*	10	*Predictions*	*Large*	*Large*	√	√	√	$\mathcal{L}_{CE}$, *Loss*	BERT-110M
FedDAT [72]	*Cross-Silos*	25	*Adapter*	*Large*	*Large*	√	–	√	$\mathcal{L}_{KL}$	ViLT-87M
FedCoLLM [68]	*Cross-Silos*	4	*Adapter*	*Large*	*Small*	√	√	–	$\mathcal{L}_{KL}$	GPT-2-774M
FedBiOT [69]	*Cross-Silos*	9	*Adapter*	*Large*	*Small*	–	√	–	$\mathcal{L}_2$, $\mathcal{L}_{KL}$	LLaMA2-7B
FedPFT [73]	*Cross-Devices*	100	*Proxy Model*	*Large*	*Small*	–	√	–	$\mathcal{L}_2$	BERT-110M
FedPT [74]	*Cross-Silos*	10	*Proxy Model*	*Large*	*Small*	–	√	–	$\mathcal{L}_{KL}$	GPT-2-1.5B

all kinds of model-interpretable training signals (such as adapters [64,68], logits [67], and proxy models [73]), which is quite different from gradient-based signals in the Fed-ST. The federated adapter learning or transfer learning has utilized diverse knowledge signals to improve the performance of the rLLM. However, the potential privacy risks associated with various training signals have not been fully explored. In particular, systematic and fair evaluation frameworks for measuring privacy risk across various signal types are still lacking. Thus, how to develop quantifiable metrics, like trustworthy FL [151], to assess the trade-offs among efficiency, model performance, and privacy for various knowledge formats in federated transfer learning is a promising research direction.

5.4 Summary of signal from representation

We summarize Fed-TL approaches using model-interpretable representation for rLLMs from model capability, communication efficiency, and privacy preservation as well. Firstly, concerning model performance, in federated prompt learning, increasing studies have shifted focus toward continuous prompts and explored multiple aggregation strategies for continuous prompts, e.g., the clustering-based and model-based federated aggregation. Secondly, to reduce the communication cost, LoRA [149] and Houlsby [118] adapters have been widely adopted in federated adapter learning to avoid full-parameters operations, enabling LLMs to enhance task-specific reasoning capabilities in a flexible plug-in style. Besides, various model-interpretable training signals have been explored for federated knowledge distillation to enable the collaborative training of large and small models. Then, in terms of privacy protection, it remains an open research question to what extent the various intermediate representations shared during federated transfer learning, particularly in Fed-KD and Fed-AL, may expose sensitive information. Finally, another important direction is to study which shared knowledge offers better trade-offs among model capability, communication cost, and privacy preservation.

6 Training signal from preference

As the observation that rLLMs are also policy maker [128,152], we have witnessed reinforcement learning being widely adopted as a key technique to enhance the reasoning capabilities of rLLMs [11]. This subsection introduces federated reinforcement learning (Fed-RL), which leverages the preferences from humans or models as the training signal to enhance reasoning capabilities of rLLMs. Next, we review the representative studies along this research topic.

From human feedback to AI feedback. ***Firstly***, in general, there are limited prior works utilizing Fed-RL to enhance rLLMs, and they primarily focus on Reinforcement Learning from Human Feedback (RLHF). Existing research in Fed-RL can be divided into two categories: (i) Policy based Fed-RL [77,79,123] and (ii) Selector based Fed-RL [78]. The former utilizes the heterogeneous client preferences as reinforcement signals. For example, in [79], the authors propose the FedRLHF framework, where FL clients first perform policy gradient updates locally, and then the server aggregates these policy parameters as FedAvg [121]. In contrast, the selector-based Fed-RL [78] builds a federated selector in which rLLMs first generates two reasoning outputs and prompts the user to select the preferred one. Then, the user's preference signal can be the feedback for generating the better reasoning outputs. ***Secondly,*** in response to the high annotation cost and limited scalability of human feedback, a recent study [153] has first designed a straightforward federated reinforcement learning from AI feedback (Fed-RLAIF), where rLLMs can act as judges [129] to provide preference signals, offering a more cost-effective and scalable alternative to human-in-the-loop training. In summary, using Fed-RL to improve the test-time reasoning performance of rLLMs is an emerging research direction. Notably, with recent advancements in reinforcement learning for rLLMs [154], model-based preference signals are emerging as a promising focus for the next wave of federated learning in rLLMs, where model capability, communication cost, and privacy preservation remain largely underexplored.

7 Open platforms and applications

7.1 Open-source FL platform for rLLMs

The open-source platforms is crucial to federated rLLMs, as they provide fundamental tools to support various research and application scenarios. Next, we review representative platforms (as in Fig. 9).

FATE-LLM. Building on FATE [155], FATE-LLM [156] is as the first industrial-level Fed-rLLM framework, which highlights the privacy protection. It supports various commonly used privacy preserving techniques like secure aggregation [105], differential privacy (DP) [107], and secure multi-part computation (SMC) [110], which preserve data privacy for both training and inference phases.

FederatedScope-LLM (FS-LLM). FS-LLM [157] is an open sourced comprehensive toolkit designed to fine-tune rLLMs in the FL scenario. FS-LLM offers a diverse set of federated fine-tuning datasets across multiple domains, along with corresponding evaluation tasks, to support a comprehensive benchmarking pipeline. It offers a variety of fine-tuning algorithms through unified and flexible interfaces, and supports multiple federated training modes, including simulated, distributed, and clustered modes. It is also equipped with a range of acceleration and resource-efficient operators as well as parallelization operators, which can be seamlessly integrated with advanced algorithms such as personalization or hyperparameter optimization.

Shepherd. *Shepherd* is the first open-source platform specifically designed for Federated instruction tuning (Fed-IT) [34]. The goal of Fed-IT is to leverage federated learning to address issues in data acquisition and privacy during the instruction tuning of rLLMs.

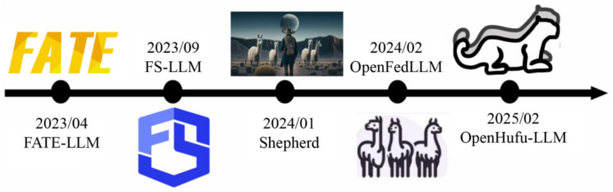


Fig. 9 Timeline of Federated rLLMs Platforms

Shepherd is in a decentralized architecture, where the FL clients first update the model adapters based on local instructions, and then the FL server is responsible for the scheduling and aggregation. It encompasses key functions (e.g., data allocation, client scheduling, and model aggregation) and is compatible with multiple rLLMs.

OpenFedLLM. The above platforms primarily concentrate on Supervised Fine-Tuning (SFT) [131]. In contrast, OpenFedLLM [158] stands out as a full-process platform. Notably, it is the first platform to support RLHF [131] within federated rLLMs. The architecture of OpenFedLLM predominantly follows a centralized model, where a central server orchestrates the training process across clients. Like FS-LLM, OpenFedLLM is a benchmark pipeline equipped with various well-defined datasets and a comprehensive set of metrics. It uses communication compression techniques such as quantization [159] for performance optimisation.

OpenHufu-LLM. OpenHufu-LLM [160–162] is an new open-source framework designed for federated fine-tuning of rLLMs. Its system architecture integrates four core components: a data access module, a federated communication module, a model fine-tuning module, and a performance evaluation module. The key strengths of OpenHufu-LLM lie in its system-level performance. It demonstrates notable communication efficiency and scalability, supporting distributed multi-GPU fine-tuning and compatibility with mainstream rLLMs. The communication module is implemented using *gRPC*, enabling the high throughput communication for large-scale parameter exchange in the federated training of rLLMs.

Summary. As shown in Table 9, all surveyed platforms support adapter learning and prompt learning. However, when it comes to operator optimization and multi-GPU parallel training, both essential for large-scale applications, only limited platforms meet these requirements. Furthermore, in terms of heterogeneity support, FATE-LLM currently supports heterogeneous model architectures, whereas most other open-source platforms primarily address data heterogeneity. In summary, existing open-source federated LLM platforms remain in the early stages of development and have yet to offer comprehensive support for reasoning-oriented techniques and large-scale industrial applications.

7.2 Typical applications of federated rLLMs

In this subsection, we discuss two representative real-world applications of federated rLLMs including Software Engineering and Medical Healthcare.

Federated LLMs for software engineering. In software engineering, the rapid evolution of technologies and projects has introduced numerous challenges for tasks such as code translation and code review, especially when source codes in various languages or formats is distributed across private repositories. For example, Kumar et al. [101] propose a federated rLLMs approach for code translation, enabling clients to jointly train a code translator without sharing sensitive data. They mainly focus on two task, i.e., *code translation* and *code review*. We review each of these tasks as follows. *(i) code translation*: Due to the proprietary nature of code, enterprises often train exclusive rLLMs locally. Experimental results in [101] demonstrate that the proposed federated approach significantly outperforms individual client models on C#-to-Java and Java-to-C# translation tasks, achieving over a 40% improvement in CodeBLEU scores. *(ii) code review*: Traditional human-driven code review processes are often time-consuming and labor-intensive. Kumar et al. [57] fine-tune the LLaMA-3-8B model to develop a multi-task rLLM for code review, encompassing tasks such as review necessity prediction, comment generation, and code refinement. Their findings show that federated rLLMs offer clear advantages for federated clients with varying code quality, particularly in code refinement. Moreover, models trained cumulatively across tasks outperforms individually fine-tuned single-task models.

Federated rLLMs for medical healthcare. In medical field, the traditional centralized paradigm for training and deploying rLLMs faces significant challenges such as elated to data privacy, communication cost, and scalability [163]. Therefore, federated rLLMs have emerged as a promising solution. We review two representative applications of federated rLLMs in the medical field: (i) medical text generation and (ii) disease diagnosis. Firstly, for medical text generation, Jung et al. [15] proposed a federated LLM framework that integrates client-specific retrieval-augmented generation (RAG) systems. This framework enables distributed retrieval and generation over local datasets while preserving the privacy of sensitive medical information. Secondly, for disease diagnosis, Liu et al. [100] introduced the FedARC, a personalized FL approach designed for multi-center tuberculosis diagnosis. The FedARC addresses challenges such as data heterogeneity and local data scarcity by leveraging adaptive regularization and model contrastive learning [164]. Their experimental results on five public chest X-ray datasets demonstrate that the federated rLLMs can significantly improve diagnostic performance, providing a accurate and efficient solution.

Table 9 Summary of the representative Fed-LLM platforms

	Privacy Tech.	Multi-GPU	Ops Optimization	Instr.-Tuning	Adapter-Tuning	Prompt-Learning	RLHF	Benchmark
FATE-LLM [156]	*DP, SMC etc.*	√	–	×	√	√	×	×
FS-LLM [157]	*DP etc.*	√	√	√	√	√	×	√
Shepherd [34]	–	×	–	√	√	√	×	×
OpenFedLLM [158]	–	×	–	√	√	√	√	√
OpenHufu-LLM [162]	*DP, SMC etc.*	√	√	√	√	√	×	×

8 Future directions and challenges

Prior studies in federated rLLMs have made significant progress by successfully utilizing various training signal formats to enhance reasoning capabilities of language models. Nevertheless, this rapidly evolving field still offers abundant research opportunities. In the following, we highlight two promising research directions and associated challenges from the perspective of training signals fed into rLLMs, including: (i) Federated RL-Enhanced rLLMs and (ii) Federated RAG-Enhanced rLLMs.

8.1 Federated RL-enhanced rLLMs

In this subsection, we envision a paradigm shift in federated reinforcement learning enhanced rLLMs, where an alternative format of training signal, namely *outcome signal*, is further exploited to enhance reasoning capabilities. Outcome signals, such as evaluation scores, are typically scalar values and introduce minimal communication overhead. Thus, it should focus more on model capability and data privacy in this research direction. Specifically, we first introduce the shift in training signals of the rLLMs and then explore the opportunities and challenges that arise across multiple outcome owners.

Training signal: from preference to outcome. Prior works [4,128,152] have already viewed the rLLM as a policy π_Θ and the next-token-prediction problem in language model can be modeled as the sequence-decision problem in reinforcement learning, where decision trajectory can be formalized as,

$$\begin{aligned} P(y|X) &= \prod_{t=1}^{T_r} P(y_t \mid X, y_1, y_2, \ldots, y_{t-1}) \\ &= \underbrace{\prod_{t=1}^{k} \pi_\Theta(r_t \mid X, r_{<t})}_{\textbf{Reasoning Decisions}} \cdot \underbrace{\prod_{t=k+1}^{k+T_r} \pi_\Theta(y_{t-k} \mid X, r_{<k}, y_{<t})}_{\textbf{Answer Decisions}}, \end{aligned} \tag{20}$$

where π_Θ is the rLLMs-based decision policy to generate a sequence of token. In such cases, various forms of sparse training signals can be utilized, such as human preference feedback [128]. Recently, the remarkable success of DeepSeek-R1 [11] etc., have shown that task-specific outcome-based scoring signals can be effectively leveraged through RL to improve the reasoning performance of rLLMs. For example, the final answers generated by rLLMs can be evaluated against known correct solutions in mathematical tasks, or code outputs can be scored based on predefined test cases. Furthermore, rLLMs can also act as judges to evaluate the outputs [129]. On the one hand, in this context, various application domains can serve as judges by providing valuable outcome-based training signals, making it feasible to apply Fed-RL across a wide range of scenarios, particularly in the privacy-sensitive applications such as healthcare. On the other hand, outcome scores typically have less privacy than other training signals (e.g., raw data or model parameters), so we believe that Fed-RL will have growing impacts and envision the opportunities and challenges in this research direction.

Firstly, FL clients may produce outcomes based on diverse and potentially conflicting criteria, resulting in inherently heterogeneous outcome-score signals. Thus, a crucial challenge in this context is how to handle such heterogeneity within Fed-RL frameworks, especially for scenarios where FL clients have non-uniform score or reward structures.

Secondly, a commonly adopted approach in RL-based rLLMs is *rejection sampling*, which filters for high-quality reasoning trajectories, serving as a valuable form of reasoning signal [11]. Therefore, an important question is whether the federated rejection sampling mechanism can be developed to leverage private reasoning data across data owners in the training of federated RL enhanced rLLMs.

Finally, existing privacy-preserving techniques, such as *secure aggregation* provide an efficient mechanism for previous federated learning algorithms [105]. Accordingly, an important research question is whether the widely-adopted RL algorithms for rLLMs, such as GRPO [11], can be reformulated into Fed-RL fashion solely using secure aggregation in the context of outcome-based signals.

8.2 Federated RAG-enhanced rLLMs

In this subsection, we introduce federated retrieval-augmented generation (Fed-RAG) of rLLMs, which exploit the external data or knowledge across various data owners in a federated fashion. Specifically, we first introduce the basic concepts and workflow and then envision the future directions for the key steps and knowledge formats of federated RAG.

External knowledge: from public to private. With in-context learning capabilities, rLLMs incorporate various formats of knowledge (e.g., prompts or embeddings) in reasoning process, enabling them to further exploit the external knowledge without requiring additional training. In the seminar work [165], Lewis et al. introduced *Retrieval-Augmented Generation*, a generation paradigm that combines language model and external knowledge for next-token-prediction, which can be formalized as,

$$P(y \mid X) = \sum_{z \in Z} P_\eta(z|X) \prod_{t=1}^{T} P(y_t \mid X, z, y_{<t}), \tag{21}$$

where Z is the retrieved external knowledge and $P_\eta(\cdot|X)$ denotes the generation model based on input tokens X and external knowledge Z. It is evident that we can achieve a same retrieval-augmented framework for reasoning by just replacing the LM in Eq. (21) with rLLMs in Eq. (3) in Section 2. In practical scenarios, e.g., healthcare [166], external clinical knowledge with sensitive information is typically distributed across various data owners. Thus, researchers have extended the concepts of RAG [165] to Fed-RAG [167–169], which allows the rLLMs to further utilize the private data or knowledge in reasoning process in a privacy-preserving manner. We envision a workflow for Fed-RAG enhanced rLLMs.

Workflow of Fed-RAG. As shown in Fig. 10, there are five steps in a typical workflow of Fed-RAG. $Step$-①: When receiving a query from a user query, the rLLM encodes the query into embeddings and sends them to the *Retrieval Router*. $Step$-②: Then, the *Retrieval Router* selects the appropriate data sources based on the semantic

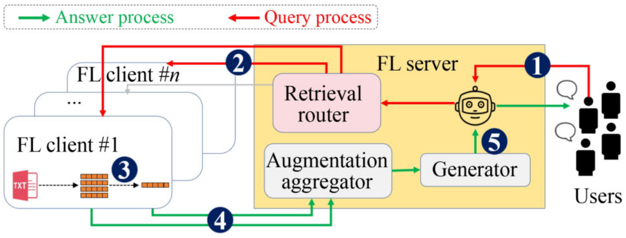


Fig. 10 The key steps in federated RAG of `rLLMs`

content of the query. It chooses data sources most likely to contain highly relevant external knowledge for the user query to avoid a naive router requiring all FL clients. *Step*-③: The selected FL clients perform local retrieval (e.g., approximate nearest neighbour search) over their private databases and return the retrieved results to the FL server, where we can employ high dimensional vector retrieval methods such as FAISS [170] or HNSW [171] on the FL client side for local retrieval. *Step*-④: The FL server then aggregates the retrieval results from all participating FL clients. In the aggregation phase of federated RAG, the FL server merges the retrieval results from previously selected FL clients for the subsequent reasoning tokens generation. *Step*-⑤: Finally, the `rLLM` utilizes the aggregated information as external knowledge to generate reasoning paths and the final answers. We envision the research questions below.

Firstly, as previously discussed, `rLLMs` are inherently flexible and trained across diverse training signals. Existing Fed-RAG solutions [167–169] predominantly operate on prompts or embeddings, and whether other formats of knowledge can be utilized reasoning across through Fed-RAG is a non-trivial problem. Furthermore, another key open problem in Fed-RAG is whether and how heterogeneous forms of knowledge (e.g., prompts, embeddings, or proxy models) can be jointly fused as external knowledge.

Secondly, for task-specific reasoning process, `rLLMs` may have different requirements for external knowledge, such as diverse viewpoints, factual correctness, or efficient response, which highlights the importance of retrieval routers and raise an important research question, i.e., can we design query-aware and intent-driven retrieval routing or indexing mechanism across FL clients as Fed-RAG infrastructures to fill the reasoning requirements?

Finally, existing Fed-RAG frameworks [167–169] mainly adopts naive aggregation strategies, such as concatenation [15] or Top-k selection [167], which not only overlooks the semantic dependencies among retrieved external knowledge but also inadvertently leaks the privacy. Thus, it is a crucial research problem whether we can design an effective FL aggregation mechanism in a *privacy-for-free* fashion.

9 Conclusion

In this survey, we provide a comprehensive overview of existing research on federated reasoning large language models (federated `rLLMs`). We propose a novel taxonomy grounded the nature of training signals, encompassing signals derived from raw data, model-interpretable representations, and preference feedback. Within the proposed taxonomy, we summarize the primary federated learning techniques and highlight the emerging trends that align with the unique features of `rLLMs`. We then review open-source platforms for federated `rLLMs` and present two representative real-world applications. Finally, we envision future research opportunities and challenges in federated `rLLMs` along two promising directions, i.e., extending the forms of training signals and leveraging externally sourced knowledge.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (NSFC) (Grant Nos. 62425202, U21A20516, 62336003), the Beijing Natural Science Foundation (Z230001), the Fundamental Research Funds for the Central Universities (No. JK2024-03), the Didi Collaborative Research Program and the State Key Laboratory of Complex & Critical Software Environment (SKLCCSE). Zimu Zhou's research is supported by Chow Sang Sang Group Research Fund (No. 9229139).

Competing interests

Yongxin TONG is an Action Editor of the journal and a co-author of this article. To minimize bias, he was excluded from all editorial decision-making related to the acceptance of this article for publication. The remaining authors declare no conflict of interest.

Open Access

References

[1] Proudfoot M, Lacey A R. The Routledge Dictionary of Philosophy. 3rd ed. London: Routledge, 2009

[2] Audi R. The Cambridge Dictionary of Philosophy. Cambridge: Cambridge University Press, 1999

[3] Johnson-Laird P N. Deductive reasoning. Annual Review of Psychology, 1999, 50(1): 109−135

[4] Xu F, Hao Q, Zong Z, Wang J, Zhang Y, Wang J, Lan X, Gong J, Ouyang T, Meng F, Shao C, Yan Y, Yang Q, Song Y, Ren S, Hu X, Li Y, Feng J, Gao C, Li Y. Towards large reasoning models: a survey of reinforced reasoning with large language models. 2025, arXiv preprint arXiv: 2501.09686

[5] Plaat A, Wong A, Verberne S, Broekens J, van Stein N, Back T. Reasoning with large language models, a survey. 2024, arXiv preprint arXiv: 2407.11511

[6] Bandyopadhyay D, Bhattacharjee S, Ekbal A. Thinking machines: a survey of LLM based reasoning strategies. 2025, arXiv preprint arXiv: 2503.10814

[7] Huang J, Chang K C C. Towards reasoning in large language models: a survey. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023. 2023, 1049−1065

[8] Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, Chi E H, Le Q V, Zhou D. Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1800

[9] Zhang Z, Zhang A, Li M, Smola A. Automatic chain of thought prompting in large language models. In: Proceedings of the 11th International Conference on Learning Representations. 2023

[10] Yao S, Yu D, Zhao J, Shafran I, Griffiths T L, Cao Y, Narasimhan K. Tree of thoughts: deliberate problem solving with large language models. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 517

[11] Guo D, Yang D, Zhang H, Song J, Zhang R, et al. Deepseek-R1: incentivizing reasoning capability in LLMs via reinforcement learning. 2025, arXiv preprint arXiv: 2501.12948

[12] OpenAI. OpenAI Gpt-4.5 system card. See Openai.com/index/gpt-4–5-system-card/ website, 2025

[13] Anthropic. Claude 3.7 sonnet system card, 2025

[14] Team Q. Qwen3 technical report, 2025

[15] Jung J, Jeong H, Huh E N. Federated learning and RAG integration: a scalable approach for medical large language models. In: Proceedings of 2025 International Conference on Artificial Intelligence in Information and Communication. 2025, 968−973

[16] Kumar J, Chimalakonda S. Code summarization without direct access to code - towards exploring federated LLMs for software engineering. In: Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering. 2024, 100−109

[17] Regulation P. General data protection regulation. Intouch, 2018, 25: 1–5

[18] Kairouz P, McMahan H B, Avent B, Bellet A, Bennis M, Bhagoji A N, Bonawitz K, Charles Z, Cormode G, Cummings R, D'Oliveira R G L, Eichner H, Rouayheb S E, Evans D, Gardner J, Garrett Z, Gascón A, Ghazi B, Gibbons P B, Gruteser M, Harchaoui Z, He C, He L, Huo Z, Hutchinson B, Hsu J, Jaggi M, Javidi T, Joshi G, Khodak M, Konecný J, Korolova A, Koushanfar F, Koyejo S, Lepoint T, Liu Y, Mittal P, Mohri M, Nock R, Özgür A, Pagh R, Qi H, Ramage D, Raskar R, Raykova M, Song D, Song W, Stich S U, Sun Z, Suresh A T, Tramèr F, Vepakomma P, Wang J, Xiong L, Xu Z, Yang Q, Yu F X, Yu H, Zhao S. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 2021, 14(1-2): 1−210

[19] Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 12

[20] Liu F, Zheng Z, Shi Y, Tong Y, Zhang Y. A survey on federated learning: a perspective from multi-party computation. Frontiers of Computer Science, 2024, 18(1): 181336

[21] Wei S, Zhang Y, Zhou Z, Zhang T, Xu K. FedSM: a practical federated shared mobility system. Proceedings of the VLDB Endowment, 2024, 17(12): 4445−4448

[22] Song T, Tong Y, Wei S. Profit allocation for federated learning. In: Proceedings of 2019 IEEE International Conference on Big Data. 2019, 2577−2586

[23] Tong Y, Zeng Y, Zhou Z, Liu B, Shi Y, Li S, Xu K, Lv W. Federated computing: query, learning, and beyond. IEEE Data Engineering Bulletin, 2023, 46(1): 9−26

[24] Zhang J, Yang H F, Li A, Guo X, Wang P, Wang H, Chen Y, Li H. MLLM-FL: multimodal large language model assisted federated learning on heterogeneous and long-tailed data. 2024, arXiv preprint arXiv: 2409.06067

[25] Yang H, Sui M, Liu S, Qian X, Zhang Z, Liu B. Research on key technologies for cross-cloud federated training of large language models. Academic Journal of Computing & Information Science, 2024, 7(11): 42−49

[26] Sani L, Iacob A, Cao Z, Marino B, Gao Y, Paulik T, Zhao W, Shen W F, Aleksandrov P, Qiu X, Lane N D. The future of large language model pre-training is federated. In: Proceedings of the International Workshop on Federated Foundation Models in Conjunction with NeurIPS. 2024

[27] Shu Y, Hu W, Ng S K, Low B K H, Yu F R. Ferret: federated full-parameter tuning at scale for large language models. In: Proceedings of the International Workshop on Federated Foundation Models in Conjunction with NeurIPS. 2024

[28] Cai Z, Chen H, Zhu G. FeedSign: robust full-parameter federated fine-tuning of large models with extremely low communication overhead of one bit. 2025, arXiv preprint arXiv: 2501.17610

[29] Zhang Z, Cai D, Zhang Y, Xu M, Wang S, Zhou A. FedRDMA: communication-efficient cross-silo federated LLM via chunked RDMA transmission. In: Proceedings of the 4th Workshop on Machine Learning and Systems. 2024, 126−133

[30] Xu M, Cai D, Wu Y, Li X, Wang S. FwdLLM: efficient federated finetuning of large language models with perturbed inferences. In: Proceedings of 2024 USENIX Conference on USENIX Annual Technical Conference. 2024, 579−596

[31] Wang L, Wang Z, Tang X. Save it all: enabling full parameter tuning for federated large language models via cycle black gradient descent. 2024, arXiv preprint arXiv: 2406.11187

[32] Ling Z, Chen D, Yao L, Li Y, Shen Y. 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. 2024, 1827−1838

[33] Qin Z, Chen D, Qian B, Ding B, Li Y, Deng S. Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 1686

[34] Zhang J, Vahidian S, Kuo M, Li C, Zhang R, Yu T, Wang G, Chen Y. Towards building the federatedgpt: federated instruction tuning. In: Proceedings of 2024 IEEE International Conference on Acoustics, Speech and Signal Processing. 2024, 6915−6919

[35] Du Y, Ye R, Yuchi F, Zhao W, Qu J, Wang Y, Chen S. Data quality control in federated instruction-tuning of large language models. 2024, arXiv preprint arXiv: 2410.11540

[36] Qin Z, Wu Z, He B, Deng S. Federated data-efficient instruction tuning for large language models. 2024, arXiv preprint arXiv: 2410.10926

[37] Ye R, Ge R, Yuchi F, Chai J, Wang Y, Chen S. Leveraging

unstructured text data for federated instruction tuning of large language models. In: Proceedings of the Federated Learning in the Age of Foundation Models - FL 2024 International Workshops. 2025, 119−131

[38] Zhang Z, Zhang J, Huang J, Qu L, Zhang H, Wang Q, Zhou X, Xu Z. FewFedPIT: towards privacy-preserving and few-shot federated instruction tuning. 2024, arXiv preprint arXiv: 2403.06131

[39] Liu X, Pang T, Fan C. Federated prompting and chain-of-thought reasoning for improving LLMs answering. In: Proceedings of the 16th International Conference on Knowledge Science, Engineering and Management. 2023, 3−11

[40] Seo J, Zhang N, Rong C. Flexible and secure code deployment in federated learning using large language models: prompt engineering to enhance malicious code detection. In: Proceedings of 2023 IEEE International Conference on Cloud Computing Technology and Science. 2023, 341−349

[41] Wu P, Li K, Nan J, Wang F. Federated in-context LLM agent learning. 2024, arXiv preprint arXiv: 2412.08054

[42] Pan B, Huang W, Shi Y. Federated learning from vision-language foundation models: theoretical analysis and method. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 30590−30623

[43] Su S, Yang M, Li B, Xue X. Federated adaptive prompt tuning for multi-domain collaborative learning. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 15117−15125

[44] Che T, Liu J, Zhou Y, Ren J, Zhou J, Sheng V, Dai H, Dou D. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In: Proceedings of 2023 Conference on Empirical Methods in Natural Language Processing. 2023, 7871−7888

[45] Guo T, Guo S, Wang J, Tang X, Xu W. PromptFL: let federated participants cooperatively learn prompts instead of models - federated learning in age of foundation model. IEEE Transactions on Mobile Computing, 2024, 23(5): 5179−5194

[46] Wu J, Chen S, Yang Y, Li Y, Hou S, Jing R, Wang Z, Chen W, Tian Z. FedDTPT: federated discrete and transferable prompt tuning for black-box large language models. 2024, arXiv preprint arXiv: 2411.00985

[47] Gong S, Cui C, Zhang C, Wang W, Nie X, Zhu L. Federated domain generalization via prompt learning and aggregation. 2024, arXiv preprint arXiv: 2411.10063

[48] Sun J, Xu Z, Yin H, Yang D, Xu D, Liu Y, Du Z, Chen Y, Roth H R. FedBPT: efficient federated black-box prompt tuning for large language models. In: Proceedings of the 41st International Conference on Machine Learning. 2024

[49] Zhuang W, Chen C, Lyu L. When foundation model meets federated learning: motivations, challenges, and future directions. 2023, arXiv preprint arXiv: 2306.15546

[50] Yao Y, Zhang J, Wu J, Huang C, Xia Y, Yu T, Zhang R, Kim S, Rossi R, Li A, Yao L, McAuley J, Chen Y, Joe-Wong C. Federated large language models: current progress and future directions. 2024, arXiv preprint arXiv: 2409.15723

[51] Ren C, Yu H, Peng H, Tang X, Zhao B, Yi L, Tan A Z, Gao Y, Li A, Li X, Li Z, Yang Q. Advances and open challenges in federated foundation models. IEEE Communications Surveys & Tutorials, 2025

[52] Fan T, Gu H, Cao X, Chan C S, Chen Q, Chen Y, Feng Y, Gu Y, Geng J, Luo B, Liu S, Ong W K, Ren C, Shao J, Sun C, Tang X, Tae H X, Tong Y, Wei S, Wu F, Xi W, Xu M, Yang H, Yang X, Yan J, Yu H, Yu H, Zhang T, Zhang Y, Zhang X, Zheng Z, Fan L, Yang Q. Ten challenging problems in federated foundation models. IEEE Transactions on Knowledge and Data Engineering, 2025

[53] Woisetschläger H, Erben A, Wang S, Mayer R, Jacobsen H A. A survey on efficient federated learning methods for foundation model training. In: Proceedings of the 33rd International Joint Conference on Artificial Intelligence. 2024, 8317−8325

[54] Hu J, Wang D, Wang Z, Pang X, Xu H, Ren J, Ren K. Federated large language model: solutions, challenges and future directions. IEEE Wireless Communications, 2024

[55] Li S, Ye F, Fang M, Zhao J, Chan Y H, Ngai E C H, Voigt T. Synergizing foundation models and federated learning: a survey. 2024, arXiv preprint arXiv: 2406.12844

[56] Wang Z, Shen Z, He Y, Sun G, Wang H, Lyu L, Li A. FLoRA: federated fine-tuning large language models with heterogeneous low-rank adaptations. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 22513−22533

[57] Kumar J, Chimalakonda S. Code review automation via multi-task federated LLM — an empirical study. 2024, arXiv preprint arXiv: 2412.15676

[58] Singhal R, Ponkshe K, Vartak R, Varshney L R, Vepakomma P. Fed-SB: a silver bullet for extreme communication efficiency and performance in (private) federated LoRA fine-tuning. 2025, arXiv preprint arXiv: 2502.15436

[59] Tastan N, Nandakumar K. A framework for double-blind federated adaptation of foundation models. 2025, arXiv preprint arXiv: 2502.01289

[60] Saha P, Mishra D, Wagner F, Kamnitsas K, Noble J A. FedPIA−permuting and integrating adapters leveraging Wasserstein barycenters for finetuning foundation models in multi-modal federated learning. In: Proceedings of the 39th AAAI Conference on Artificial Intelligence. 2025, 20228−20236

[61] Sun Y, Li Z, Li Y, Ding B. Improving LoRA in privacy-preserving federated learning. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[62] Yang Y, Long G, Shen T, Jiang J, Blumenstein M. Dual-personalizing adapter for federated foundation models. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 39409−39433

[63] Liu J, Ren J, Jin R, Zhang Z, Zhou Y, Valduriez P, Dou D. Fisher information-based efficient curriculum federated learning with large language models. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 10497−10523

[64] Zhao Q, Qu C, Chen C, Fan M, Wang Y. FedMCP: parameter-efficient federated learning with model-contrastive personalization. In: Proceedings of 2024 IEEE 30th International Conference on Parallel and Distributed Systems. 2024, 246−253

[65] Wu P, Li K, Wang T, Dong Y, Leung V C M, Wang F. FedFMSL: federated learning of foundation models with sparsely activated LoRA. IEEE Transactions on Mobile Computing, 2024, 23(12): 15167−15181

[66] Dong C, Xie Y, Ding B, Shen Y, Li Y. Tunable soft prompts are messengers in federated learning. In: Proceedings of the Findings of the

Association for Computational Linguistics: EMNLP 2023. 2023, 14665−14675

[67] Fan T, Ma G, Kang Y, Gu H, Song Y, Fan L, Chen K, Yang Q. FedMKT: federated mutual knowledge transfer for large and small language models. In: Proceedings of the 31st International Conference on Computational Linguistics. 2025, 243−255

[68] Fan T, Kang Y, Ma G, Fan L, Chen K, Yang Q. FedCoLLM: a parameter-efficient federated co-tuning framework for large and small language models. 2024, arXiv preprint arXiv: 2411.11707

[69] Wu F, Li Z, Li Y, Ding B, Gao J. 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. 2024, 3345−3355

[70] Fan T, Ma G, Song Y, Fan L, Chen K, Yang Q. PPC-GPT: federated task-specific compression of large language models via pruning and chain-of-thought distillation. 2025, arXiv preprint arXiv: 2502.15857

[71] Ma X, Liu J, Wang J, Zhang X. FedID: federated interactive distillation for large-scale pretraining language models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2023, 8566−8577

[72] Chen H, Zhang Y, Krompass D, Gu J, Tresp V. FedDAT: an approach for foundation model finetuning in multi-modal heterogeneous federated learning. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 11285−11293

[73] Peng Z, Fan X, Chen Y, Wang Z, Pan S, Wen C, Zhang R, Wang C. FedPFT: federated proxy fine-tuning of foundation models. In: Proceedings of the 33rd International Joint Conference on Artificial Intelligence. 2024, 4806−4814

[74] Gao Z, Zhang Y, Zhang Z, Gong Y, Guo Y. FedPT: federated proxy-tuning of large language models on resource-constrained edge devices. 2024, arXiv preprint arXiv: 2410.00362

[75] Wu X, Lin W Y, Willmott D, Condessa F, Huang Y, Li Z, Ganesh M R. Leveraging foundation models to improve lightweight clients in federated learning. 2023, arXiv preprint arXiv: 2311.08479

[76] Qi J, Zhou Q, Lei L, Zheng K. Federated reinforcement learning: techniques, applications, and open challenges. 2021, arXiv preprint arXiv: 2108.11887

[77] Srewa M, Zhao T, Elmalaki S. PluralLLM: pluralistic alignment in LLMS via federated learning. In: Proceedings of the 3rd International Workshop on Human-Centered Sensing, Modeling, and Intelligent Systems. 2025, 64−69

[78] Wu F, Liu X, Wang H, Wang X, Su L, Gao J. Towards federated RLHF with aggregated client preference for LLMs. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[79] Fan F X, Tan C, Ong Y S, Wattenhofer R, Ooi W T. FedRLHF: a convergence-guaranteed federated framework for privacy-preserving and personalized RLHF. 2024, arXiv preprint arXiv: 2412.15538

[80] Spadea F, Seneviratne O. Federated fine-tuning of large language models: Kahneman-Tversky vs. direct preference optimization. 2025, arXiv preprint arXiv: 2502.14187

[81] Zhao W X, Zhou K, Li J, Tang T, Wang X, Hou Y, Min Y, Zhang B, Zhang J, Dong Z, Du Y, Yang C, Chen Y, Chen Z, Jiang J, Ren R, Li Y, Tang X, Liu Z, Liu P, Nie J Y, Wen J R. A survey of large language models. 2023, arXiv preprint arXiv: 2303.18223

[82] El-Kassas W S, Salama C R, Rafea A A, Mohamed H K. Automatic text summarization: a comprehensive survey. Expert Systems with Applications, 2021, 165: 113679

[83] NLLB Team. Scaling neural machine translation to 200 languages. Nature, 2024, 630(8018): 841−846

[84] Su Y, Lan T, Wang Y, Yogatama D, Kong L, Collier N. A contrastive framework for neural text generation. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1566

[85] Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, et al. GPT-4 technical report. 2023, arXiv preprint arXiv: 2303.08774

[86] Anil R, Dai A M, Firat O, Johnson M, Lepikhin D, et al. PaLM 2 technical report. 2023, arXiv preprint arXiv: 2305.10403

[87] Touvron H, Martin L, Stone K, Albert P, Almahairi A, et al. Llama 2: open foundation and fine-tuned chat models. 2023, arXiv preprint arXiv: 2307.09288

[88] Wei J, Tay Y, Bommasani R, Raffel C, Zoph B, Borgeaud S, Yogatama D, Bosma M, Zhou D, Metzler D, Chi E H, Hashimoto T, Vinyals O, Liang P, Dean J, Fedus W. Emergent abilities of large language models. Transactions on Machine Learning Research, 2022, 2022

[89] Kojima T, Gu S S, Reid M, Matsuo Y, Iwasawa Y. Large language models are zero-shot reasoners. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1613

[90] Yu F, Zhang H, Tiwari P, Wang B. Natural language reasoning, a survey. ACM Computing Surveys, 2024, 56(12): 304

[91] Zhu D, Wei X, Zhao G, Wu W, Zou H, Ran J, Wang X, Sun L, Zhang X, Li S. Chain-of-thought matters: improving long-context language models with reasoning path supervision. 2025, arXiv preprint arXiv: 2502.20790

[92] Zhang X, Wang D, Dou L, Zhu Q, Che W. A survey of table reasoning with large language models. Frontiers of Computer Science, 2025, 19(9): 199348

[93] Yang A, Zhang B, Hui B, Gao B, Yu B, Li C, Liu D, Tu J, Zhou J, Lin J, Lu K, Xue M, Lin R, Liu T, Ren X, Zhang Z. Qwen2.5-math technical report: toward mathematical expert model via self-improvement. 2024, arXiv preprint arXiv: 2409.12122

[94] Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, et al. The llama 3 herd of models. 2024, arXiv preprint arXiv: 2407.21783

[95] Guo Z, Zhang R, Tong C, Zhao Z, Gao P, Li H, Heng P A. Can we generate images with CoT? Let's verify and reinforce image generation step by step. 2025, arXiv preprint arXiv: 2501.13926

[96] Besta M, Blach N, Kubicek A, Gerstenberger R, Podstawski M, Gianinazzi L, Gajda J, Lehmann T, Niewiadomski H, Nyczyk P, Hoefler T. Graph of thoughts: solving elaborate problems with large language models. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 17682−17690

[97] Cobbe K, Kosaraju V, Bavarian M, Chen M, Jun H, Kaiser L, Plappert M, Tworek J, Hilton J, Nakano R, Hesse C, Schulman J. Training verifiers to solve math word problems. 2021, arXiv preprint arXiv: 2110.14168

[98] Wang J, Wang X, Lyu L, Chen J, Ma F. FEDMEKI: a benchmark for scaling medical foundation models via federated knowledge injection. In: Proceedings of the 38th International Conference on Neural

Information Processing Systems. 2024

[99] Li X, Peng L, Wang Y P, Zhang W. Open challenges and opportunities in federated foundation models towards biomedical healthcare. BioData Mining, 2025, 18(1): 2

[100] Liu C, Luo Y, Xu Y, Du B. Foundation models matter: federated learning for multi-center tuberculosis diagnosis via adaptive regularization and model-contrastive learning. World Wide Web, 2024, 27(3): 30

[101] Kumar J, Janapati V L S, Tanguturi M R, Chimalakonda S. I can't share code, but I need translation -- an empirical study on code translation through federated LLM. 2025, arXiv preprint arXiv: 2501.05724

[102] Anonymous Hackers. Is the deep web 90% of the internet? See Anonymoushackers.net/dark-web-news/is-the-deep-web-90-of-the-internet/ website, 2025

[103] de la Torre L. A guide to the California consumer privacy act of 2018. See Papers.ssrn.com/sol3/papers.cfm?abstract_id=3275571 website, 2018

[104] Baracaldo N. Is federated learning still alive in the foundation model era? In: Proceedings of the AAAI 2024 Spring Symposium Series. 2024, 293

[105] Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan H B, Patel S, Ramage D, Segal A, Seth K. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017, 1175−1191

[106] Chua T J, Yu W, Zhao J, Lam K Y. FedPEAT: convergence of federated learning, parameter-efficient fine tuning, and emulator assisted tuning for artificial intelligence foundation models with mobile edge computing. 2023, arXiv preprint arXiv: 2310.17491

[107] Dwork C. Differential privacy. In: Proceedings of the 33rd International Colloquium on Automata, Languages and Programming. 2006, 1−12

[108] Dwork C. Differential privacy: a survey of results. In: Proceedings of the 5th International Conference on Theory and Applications of Models of Computation. 2008, 1−19

[109] Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st annual ACM Symposium on Theory of Computing. 2009, 169−178

[110] Zhao C, Zhao S, Zhao M, Chen Z, Gao C Z, Li H, Tan Y A. Secure multi-party computation: theory, practice and applications. Information Sciences, 2019, 476: 357−372

[111] Yao A C. Protocols for secure computations. In: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science. 1982, 160−164

[112] Kifer D, Machanavajjhala A. No free lunch in data privacy. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data. 2011, 193−204

[113] Yang L, Chen W, He X, Wei S, Xu Y, Zhou Z, Tong Y. FedGTP: exploiting inter-client spatial dependency in federated graph-based traffic prediction. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024, 6105−6116

[114] Sun B, Zhang H, Wu Z, Zhang Y, Li T. Adaptive spatiotemporal graph convolutional networks for motor imagery classification. IEEE Signal Processing Letters, 2021, 28: 219−223

[115] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning. 2008, 160−167

[116] Xie Q, Dai Z, Hovy E, Luong M T, Le Q V. Unsupervised data augmentation for consistency training. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 525

[117] Zhou D, Bousquet O, Lal T N, Weston J, Schölkopf B. Learning with local and global consistency. In: Proceedings of the 17th International Conference on Neural Information Processing Systems. 2003, 321−328

[118] Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, De Laroussilhe Q, Gesmundo A, Attariyan M, Gelly S. Parameter-efficient transfer learning for NLP. In: Proceedings of the 36th International Conference on Machine Learning. 2019, 2790−2799

[119] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2019, arXiv preprint arXiv: 1910.01108

[120] Brown T B, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D M, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D. Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 159

[121] Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V, Kiddon C, Konečný J, Mazzocchi S, McMahan B, Van Overveldt T, Petrou D, Ramage D, Roselander J. Towards federated learning at scale: system design. In: Proceedings of the 2nd Conference on Machine Learning and Systems. 2019

[122] Guo W, Zhuang F, Zhang X, Tong Y, Dong J. A comprehensive survey of federated transfer learning: challenges, methods and applications. Frontiers of Computer Science, 2024, 18(6): 186356

[123] Jin H, Peng Y, Yang W, Wang S, Zhang Z. Federated reinforcement learning with environment heterogeneity. In: Proceedings of the 25th International Conference on Artificial Intelligence and Statistics. 2022, 18−37

[124] Robbins H, Monro S. A stochastic approximation method. The Annals of Mathematical Statistics, 1951, 22(3): 400−407

[125] Kingma D P, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations. 2015

[126] Lee G, Jeong M, Shin Y, Bae S, Yun S Y. Preservation of the global knowledge by not-true distillation in federated learning. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 2787

[127] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, 6000−6010

[128] Christiano P F, Leike J, Brown T B, Martic M, Legg S, Amodei

D. Deep reinforcement learning from human preferences. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, 4302−4310

[129] Gu J, Jiang X, Shi Z, Tan H, Zhai X, Xu C, Li W, Shen Y, Ma S, Liu H, Wang Y, Guo J. A survey on LLM-as-a-judge. 2024, arXiv preprint arXiv: 2411.15594

[130] Wen B, Zhang X. Enhancing reasoning to adapt large language models for domain-specific applications. 2025, arXiv preprint arXiv: 2502.04384

[131] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C L, Mishkin P, Zhang C, Agarwal S, Slama K, Ray A, Schulman J, Hilton J, Kelton F, Miller L, Simens M, Askell A, Welinder P, Christiano P, Leike J, Lowe R. Training language models to follow instructions with human feedback. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 2011

[132] Zhu L, Liu Z, Han S. Deep leakage from gradients. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019, 1323

[133] Wang H, Sreenivasan K, Rajput S, Vishwakarma H, Agarwal S, Sohn J Y, Lee K, Papailiopoulos D. Attack of the tails: yes, you really can backdoor federated learning. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 1348

[134] Wang F, Hugh E, Li B. More than enough is too much: adaptive defenses against gradient leakage in production federated learning. IEEE/ACM Transactions on Networking, 2024, 32(4): 3061−3075

[135] Wang Z, Du Y, Ma X, Jiang Y, Qian Z, Chen S. Federated instruction tuning of LLMs with domain coverage augmentation. 2024, arXiv preprint arXiv: 2409.20135

[136] Zhang R, Wang Y, Zhou Z, Ren Z, Tong Y, Xu K. Data source selection in federated learning: a submodular optimization approach. In: Proceedings of the 27th International Conference on Database Systems for Advanced Applications. 2022, 606−614

[137] Fu L, Zhang H, Gao G, Zhang M, Liu X. Client selection in federated learning: principles, challenges, and opportunities. IEEE Internet of Things Journal, 2023, 10(24): 21811−21819

[138] Wei S, Tong Y, Zhou Z, Song T. Efficient and fair data valuation for horizontal federated learning. In: Yang Q, Fan L, Yu H, eds. Federated Learning: Privacy and Incentive. Cham: Springer, 2020, 139−152

[139] Wei S, Tong Y, Zhou Z, He T, Xu Y. Efficient data valuation approximation in federated learning: A sampling-based approach. In: Proceeding of the 41st IEEE International Conference on Data Engineering. 2025, 2922–2934

[140] West R, Aydin R. The AI alignment paradox. Communications of the ACM, 2025, 68(3): 24−26

[141] Ye R, Chai J, Liu X, Yang Y, Wang Y, Chen S. Emerging safety attack and defense in federated instruction tuning of large language models. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[142] White J, Fu Q, Hays S, Sandborn M, Olea C, Gilbert H, Elnashar A, Spencer-Smith J, Schmidt D C. A prompt pattern catalog to enhance prompt engineering with ChatGPT. In: Proceedings of the 30th Conference on Pattern Languages of Programs. 2023, 5

[143] Weng P Y, Hoang M, Nguyen L M, Thai M T, Weng T W, Hoang T N. Probabilistic federated prompt-tuning with non-IID and imbalanced data. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024

[144] Church K W. Word2vec. Natural Language Engineering, 2017, 23(1): 155−162

[145] Lin Z, Sun Y, Shi Y, Wang X, Huang L, Shen L, Tao D. Efficient federated prompt tuning for black-box large pre-trained models. 2023, arXiv preprint arXiv: 2310.03123

[146] Igel C, Hansen N, Roth S. Covariance matrix adaptation for multi-objective optimization. Evolutionary Computation, 2007, 15(1): 1−28

[147] Zhang C, Long G, Guo H, Fang X, Song Y, Liu Z, Zhou G, Zhang Z, Liu Y, Yang B. Federated adaptation for foundation model-based recommendations. In: Proceedings of the 33rd International Joint Conference on Artificial Intelligence. 2024, 5453−5461

[148] Zhang Z, Yang Y, Dai Y, Wang Q, Yu Y, Qu L, Xu Z. FedPETuning: when federated learning meets the parameter-efficient tuning methods of pre-trained language models. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023. 2023, 9963−9977

[149] Mao Y, Ge Y, Fan Y, Xu W, Mi Y, Hu Z, Gao Y. A survey on LoRA of large language models. Frontiers of Computer Science, 2025, 19(7): 197605

[150] Gou J, Yu B, Maybank S J, Tao D. Knowledge distillation: a survey. International Journal of Computer Vision, 2021, 129(6): 1789−1819

[151] Zhang Y, Zeng D, Luo J, Fu X, Chen G, Xu Z, King I. A survey of trustworthy federated learning: issues, solutions, and challenges. ACM Transactions on Intelligent Systems and Technology, 2024, 15(6): 112

[152] Rafailov R, Sharma A, Mitchell E, Manning C D, Ermon S, Finn C. Direct preference optimization: your language model is secretly a reward model. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023

[153] Pan Y, Su Z, Wang Y, Guo S, Liu H, Li R, Wu Y. Cloud-edge collaborative large model services: challenges and solutions. IEEE Network, 2024

[154] Chen Q, Qin L, Liu J, Peng D, Guan J, Wang P, Hu M, Zhou Y, Gao T, Che W. Towards reasoning era: a survey of long chain-of-thought for reasoning large language models. 2025, arXiv preprint arXiv: 2503.09567

[155] Liu Y, Fan T, Chen T, Xu Q, Yang Q. FATE: an industrial grade platform for collaborative learning with data protection. The Journal of Machine Learning Research, 2021, 22(1): 226

[156] Fan T, Kang Y, Ma G, Chen W, Wei W, Fan L, Yang Q. FATE-LLM: a industrial grade federated learning framework for large language models. 2023, arXiv preprint arXiv: 2310.10049

[157] Kuang W, Qian B, Li Z, Chen D, Gao D, Pan X, Xie Y, Li Y, Ding B, Zhou J. FederatedScope-LLM: a comprehensive package for fine-tuning large language models in federated learning. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024, 5260−5271

[158] Ye R, Wang W, Chai J, Li D, Li Z, Xu Y, Du Y, Wang Y, Chen S. OpenFedLLM: training large language models on decentralized private data via federated learning. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024,

6137−6147

[159] Jacob B, Kligys S, Chen B, Zhu M, Tang M, Howard A, Adam H, Kalenichenko D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition. 2018, 2704−2713

[160] Tong Y, Zeng Y, Song Y, Pan X, Fan Z, Xue C, Zhou Z, Zhang X, Chen L, Xu Y, Xu K, Lv W. Hu-Fu: efficient and secure spatial queries over data federation. The VLDB Journal, 2025, 34(2): 19

[161] Tong Y, Pan X, Zeng Y, Shi Y, Xue C, Zhou Z, Zhang X, Chen L, Xu Y, Xu K, Lv W. Hu-Fu: efficient and secure spatial queries over data federation. Proceedings of the VLDB Endowment, 2022, 15(6): 1159−1172

[162] BDA-Group. OpenHuFu-LLM. See Github.com/BUAA-BDA/OpenHufu-LLM website, 2025

[163] Dong H, Xie S. Large language models (LLMs): deployment, tokenomics and sustainability. 2024, arXiv preprint arXiv: 2405.17147

[164] Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 1597−1607

[165] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih T W, Rocktäschel T, Riedel S, Kiela D. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 793

[166] Jiang E. Clinical question-answering over distributed EHR data. Massachusetts Institute of Technology, Dissertation, 2024

[167] Guerraoui R, Kermarrec A M, Petrescu D, Pires R, Randl M, de Vos M. Efficient federated search for retrieval-augmented generation. In: Proceedings of the 5th Workshop on Machine Learning and Systems. 2025, 74−81

[168] Shojaee P, Harsha S S, Luo D, Maharaj A, Yu T, Li Y. Federated retrieval augmented generation for multi-product question answering. In: Proceedings of the 31st International Conference on Computational Linguistics: Industry Track. 2025, 387−397

[169] Yang M, Xu J, Ding W, Liu Y. FedHAP: federated hashing with global prototypes for cross-silo retrieval. IEEE Transactions on Parallel and Distributed Systems, 2024, 35(4): 592−603

[170] Douze M, Guzhva A, Deng C, Johnson J, Szilvasy G, Mazaré P E, Lomeli M, Hosseini L, Jégou H. The Faiss library. 2025, arXiv preprint arXiv: 2401.08281

[171] Malkov Y A, Yashunin D A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(4): 824−836

Shuyue WEI received the BE degree in computer science and technology from Beihang University, China in 2020. He is currently working toward the PhD degree in the School of Computer Science and Engineering, Beihang University, China. His major research interests include federated large language model, federated learning and data valuation.

Yongxin TONG received the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology, China in 2014. He is currently a professor in the School of Computer Science and Engineering, Beihang University, China. His research interests include federated large language model, federated learning, spatio-temporal data analytics, and reinforcement learning. He received the championship of KDD Cup 2020 RL track.

Zimu ZHOU received the BE from the Department of Electronic Engineering, Tsinghua University, China in 2011 and the PhD from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, China in 2015. His research focuses on mobile, ubiquitous computing, and federated learning.

Yi XU received his BE, ME and PhD degrees from Beihang University, China in 2009, 2013, and 2020, respectively. He is an associate researcher in the Institute of Artificial Intelligence, Beihang University, China. His research interests include federated learning, federated foundation models, big spatio-temporal data mining, and crowd intelligence.

Jingkai GAO received the BE degree in computer science and technology from Beihang University, China in 2024. He is currently working toward the Master's degree in School of Computer Science and Engineering, Beihang University, China. His major research interests include federated large language model and spatial-temporal federated learning.

Tongyu WEI is currently an undergraduate student in Computer Science and Technology at Beijing University of Technology, China. She will pursue an ME degree at Beihang University, China. Her research interests include federated learning and the fine-tuning of large language models.

Tianran HE received the BE degree in computer science and technology from Beihang University, China in 2024. He is currently pursuing the ME degree in computer science and technology at Beihang University, China. His research interests include federated data valuation and federated learning.

Weifeng LV received the PhD degree in computer science from Beihang University, China. He is a professor, the vice president of Beihang University, China. His research interests include big spatio-temporal data analytics, foundation models, federated learning, smart city, and crowd intelligence. He is leader of Group of National Smart City Standard.