Communication-Efficient and Personalized Federated Foundation Model Fine-Tuning via Tri-Matrix Adaptation

Yongle Li, Bo Liu, Sheng Huang, Zheng Zhang, Xiao-Tong Yuan, and Richang Hong

Abstract—In federated learning, fine-tuning pre-trained foundation models poses significant challenges, particularly regarding high communication cost and suboptimal model performance due to data heterogeneity between the clients. To address these issues, this paper introduces communication-efficient federated LoRA adaption (CE-LoRA), a method that employs a tri-factorization low-rank adaptation approach with personalized model parameter aggregation. We first present a novel LoRA parameter factorization by introducing a small-size dense matrix, which can significantly reduce the communication cost and achieve comparable empirical performance than transferring the low-rank parameter matrix used by existing methods. Without violating data privacy, the server considers the client similarity in both training dataset and model parameter space, and learns personalized weights for model aggregation. Our experiments on various LLM and VLM fine-tuning tasks demonstrate that CE-LoRA not only significantly reduces communication overhead but also improves performance under not independently and identically distributed data conditions. In addition, CE-LoRA improves data privacy protection, effectively mitigating gradient-based data reconstruction attacks.

Index Terms—Pre-trained Foundation Models, Federated Learning, Low-Rank Adaptation

I. INTRODUCTION

Pre-trained foundation models (PFMs), such as RoBERTa [1], GPT [2], and LLaVA [3], [4], are renowned for their stellar performances in a wide range of tasks. With massive training data and well-designed pretraining tasks, these foundational models excel by learning generalizable features, thus enhancing their potential of generalizing to down-stream tasks. Through targeted fine-tuning, these models enhance capabilities in specific domains, such as aligning with human preferences in dialogue systems [5], optimizing recommendation algorithms [6], and improving task-specific performance in natural language processing and computer vision [7], [8].

One pressing challenge for individual clients and small businesses seeking to apply these PFMs to their specific tasks is the lack of domain-specific data for model fine-tuning. Extensive research efforts have been dedicated to optimizing the fine-tuning efficiency of such models [9], [10]. In particular, low-rank adaptation (LoRA) and its variants are
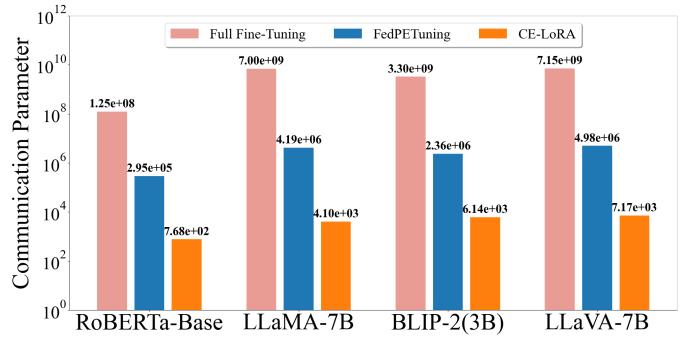
Fig. 1: Comparison of communication cost for three federated learning-based fine-tuning methods for pretrained models: transferring all model parameters, LoRA-based FL fine-tuning [12], and the proposed CE-LoRA method. The vertical axis is the parameter number needs to be transferred per iteration, on a logarithmic scale. CE-LoRA can reduce communication costs by several hundred times compared to the efficient fine-tuning of LoRA.

widely used parameter-efficient fine-tuning methods, which have demonstrated their ability to maintain robust model performance while significantly reducing the number of trainable parameters [11]. More recently, federated learning (FL)-based fine-tuning approaches have been proposed to further address the limitation of insufficient local training data by enabling collaborative model training across multiple clients. The pioneering work FedPETuning [12] exemplifies how LoRA-based parameter-efficient federated fine-tuning can drastically reduce communication overhead while maintaining adequate performance levels.

This work aims to address the following two challenges faced by federated fine-tuning of PFMs: (1) As partially illustrated by Figure 1, even with parameter-efficient fine-tuning techniques like LoRA, federated training of PFMs still requires the transmission of substantial trainable parameters, leading to high inter-client communication costs and low training efficiency; (2) Since data on involved clients tend to be not independently and identically distributed (non-IID), such type of heterogeneity between participants is among the essential technical challenges of FL. The federated average model aggregation strategy [13], [14] used in FedPETuning learns a unique global parameter update shared by all participant clients, which fails to consider the training data heterogeneity, leading to sub-optimal performance for each client.

To address these challenges, our research puts forward a communication-efficient federated LoRA adaptation approach (CE-LoRA), as well as a personalized LoRA parameter

aggregation strategy, for PFM fine-tuning. First, within the low-rank-based parameter factorization paradigm of LoRA, we present a triple-parameter factorization by introducing an additional full-rank matrix to vanilla LoRA. The introduced full-rank parameter matrix is updated by inter-client communication in order to encapsulate shared knowledge across all clients. Compared to the two low-rank components in the vanilla LoRA module, the parameter number of this full-rank matrix is much smaller; thus this design can reduce the parameter transmission overloads across clients. To achieve effective model personalization, different from the federated average model aggregation used in FedPETuning, we propose to learn a personalized model aggregation for updating each LoRA full rank component on the server. By evaluating the similarity between clients, the proposed aggregation assigns higher weights to the models of clients that are more similar. We propose a client similarity metric that jointly considers dataset similarity and model similarity. With the updated full-rank component by FL, for the two low-rank LoRA matrices, we update them with local fine-tuning. The proposed federated fine-tuning mechanism for PFMs achieves a superior balance between model fine-tuning performance, communication efficiency, and data privacy protection.

We conduct extensive experiments on federated large language model and vision-language model fine-tuning tasks on clients with heterogeneous data to evaluate the performance, communication cost, and privacy protection of CE-LoRA. Compared to state-of-the-art federated LoRA fine-tuning baseline methods, CE-LoRA improves model prediction accuracy while reducing communication overhead by hundreds of times in transmitted parameters. Moreover, CE-LoRA effectively resists gradient-based data reconstruction attacks.

II. RELATED WORK

A. Pre-trained Foundation Models

PFMs have shown broad adaptability across various tasks and domains. These models, typically trained on large-scale data using self-supervised learning, can be fine-tuned to effectively perform a wide range of downstream tasks [15]. According to the model architecture, most existing foundation models can be categorized into encoder-decoder models such as T5 [16] and BART [17], and decoder-only models like the GPT series [2], [18], [19] and LLaMA [20]. Mixture-of-experts has been used in foundation models as a flexible approach for scaling up model parameters [21], [22].

In recent years, research on PFMs has expanded from single-modal learning to multimodal learning, integrating various data types such as vision and language to improve model understanding [23], [24]. LLaVA [25] is a representative vision-language model that combines the LLaMA language model with the CLIP vision encoder and employs alignment tuning for end-to-end training, enabling the model to process image input and perform vision tasks.

Various fine-tuning approaches for foundation models, such as instruction tuning and alignment tuning [5], [10], [26], have been developed to further adapt these models to specific tasks. Low-rank adapter tuning is one such type of method that

avoids full fine-tuning by injecting trainable rank decomposition matrices into each layer of the transformer architecture [11], inspiring several recent variants. VeRA [27] employs vector-based random matrix adaptation to reduce the number of trainable parameters. AdaLoRA [28] adaptively assigns the rank of parameter matrices during fine-tuning. Q-LoRA [29] further reduces the memory overhead of LoRA by incorporating quantization techniques, enabling more efficient fine-tuning without compromising performance. Furthermore, a sparse low-rank adaptation method is proposed in [30], while mixture-of-LoRA architectures are explored in [31], [32].

B. Federated Learning

Federated learning is an emerging approach that addresses local training data insufficiency by leveraging data from multiple clients, while preserving data privacy [33]–[35]. Recent advances in federated fine-tuning of large foundation models are presented in works such as [12], [36]. In [37], [38], clustering-based and tree-based methods are proposed for federated multilingual modeling. A federated multilingual neural machine learning algorithm is introduced in [39]. Several federated prompt tuning algorithms for large models have been proposed in [40], [41]. Open-source frameworks have recently been developed to support research in federated learning for large models [42], [43]. Related topics such as algorithm privacy and efficiency are receiving increasing attention [44]–[47].

Addressing the challenge of data heterogeneity is the primary goal of personalized FL (PFL) [48]. Models such as FedProx [49] and MOON [50] alleviate the impact of data distribution heterogeneity to model with proximal regularization for global model learning. Another category of PFL methods decomposes the model based on parameter update strategies, in which global aggregation or local personalized training is applied to parameter update of different model parts [51]–[53]. Differential privacy is applied to federated LoRA learning in [54]. The communication cost of federated LoRA learning is alleviated by sparse pruning in [55]. Recent advances in personalized federated learning for foundation models involve fine-tuning dual LoRA modules [56]–[58], but a significant challenge arises from the substantial communication costs associated with aggregating LoRA parameters.

III. METHODOLOGY

A. Overview

An illustration of the proposed CE-LoRA method for federated LoRA-based PFM fine-tuning task is shown in Figure 2. The framework typically contains one server for model aggregation and multiple clients ($\{P_i\}_{i=1}^m$) where each individual client owns its private dataset and model fine-tuning infrastructure. Each client maintains a PFM whose parameters are denoted by W. The task is to jointly fine-tune the models by solving the minimization problem

$$\min_{\{L_i\}_{i=1}^m} \frac{1}{m} \sum_{i=1}^m f_i(L_i|W, D_i) \tag{1}$$

where f_i denotes the local model training loss for the i-th client P_i, which is optimized by fine-tuning the LoRA parameters
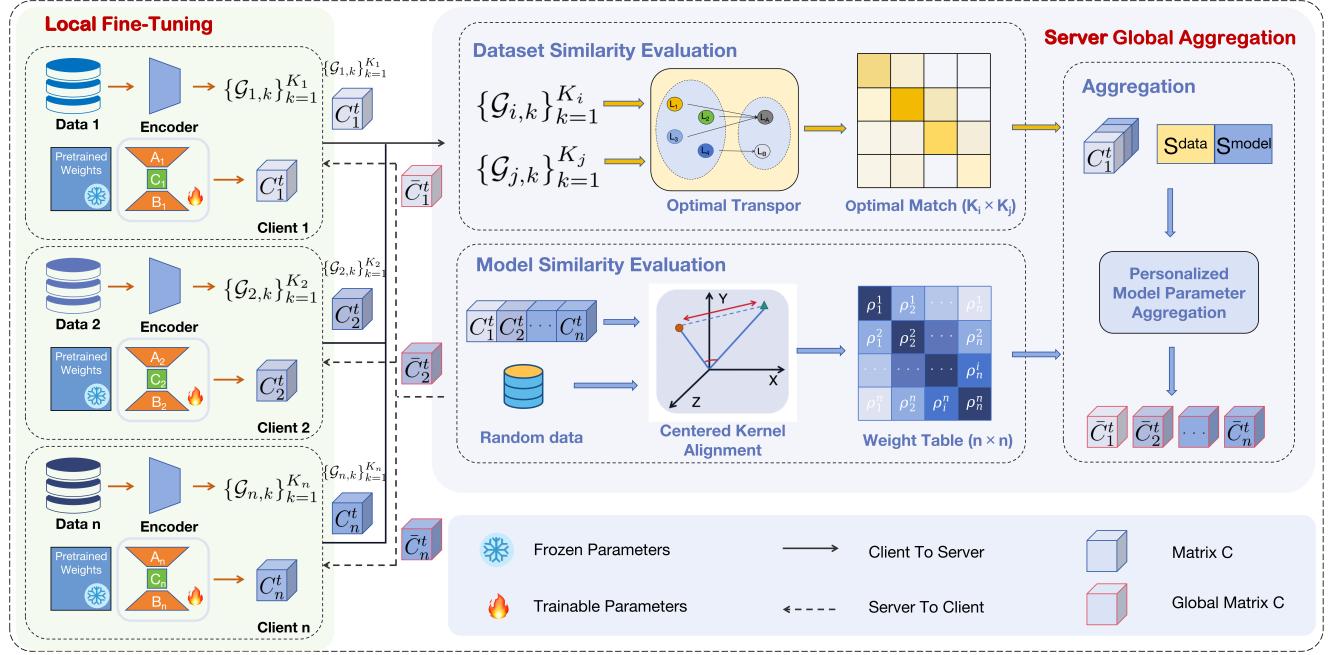
Fig. 2: Our CE-LoRA framework consists of two main components: local fine-tuning and server global aggregation. In the local fine-tuning stage, we freeze the pretrained foundation model and fine-tune it using the proposed CE-LoRA method. In addition to the low-rank matrices $A \in \mathbb{R}^{d \times r}, B \in \mathbb{R}^{r \times k}, r \ll \min(k, d)$, we introduce a full-rank matrix $C \in \mathbb{R}^{r \times r}$ that serves as the parameter matrix transmitted between clients and server. After receiving matrices $\{C_i^t\}_{i=1}^n$, the server calculates the similarity between these parameter matrices using the proposed similarity metric that jointly considers data distribution and model similarity. These client pair-wise similarities are used to derive model aggregation weights for updating $\{\bar{C}_i^t\}_{i=1}^n$.

L_i and freezing the PFM parameters; D_i is the local dataset owned by P_i.

Algorithm 1 outlines a high-level description of the proposed CE-LoRA algorithm. In the following parts we will elaborate the algorithm details, including the proposed LoRA triple factorization, personalized model parameter aggregation, local fine-tuning and model inference.

B. LoRA Triple Factorization

LoRA is among the popular parameter-efficient fine-tuning paradigms, which fine-tunes a product of two low-rank matrices as an auxiliary to the dense parameter matrix. Assume the pretrained model weight matrix is $W \in \mathbb{R}^{d \times k}$, LoRA learns two low-rank parameter matrices $A \in \mathbb{R}^{d \times r}, B \in \mathbb{R}^{r \times k}$, where $r \ll \min(d, k)$. Given input $x \in \mathbb{R}^d$, the forward pass of that layer output is obtained by

$$h = x^\top \cdot W + x^\top \cdot A \cdot B.$$

Although LoRA significantly reduces the number of trainable parameters compared to fine-tuning the entire PFM, the total number of LoRA parameters throughout the entire model can still be substantial, as partially illustrated in Figure 1. Given the iterative nature of the FL algorithm through parameter communication between the server and the clients, the communication cost of transferring LoRA parameters remains high.

In this work, we adopt a triple parameter matrix factorization, which introduces a full-rank matrix $C \in \mathbb{R}^{r \times r}$, in addition to
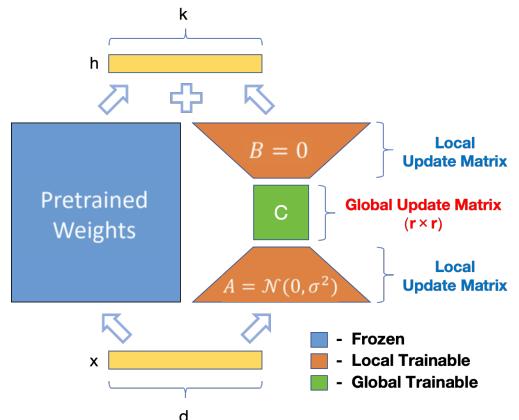
Fig. 3: Illustration of the LoRA Triple Factorization. The pretrained model is frozen during training, while the trainable LoRA is decomposed into $A \in \mathbb{R}^{r \times d}$, $B \in \mathbb{R}^{k \times r}$, and $C \in \mathbb{R}^{r \times r}$, where $r \ll \min(k, d)$. During federated learning, only C is transmitted for model parameter aggregation.

A and B for LoRA. Figure 3 illustrates the proposed LoRA triple factorization method. With the triple factorization, the LoRA forward pass becomes

$$h = x^\top \cdot W + x^\top \cdot A \cdot C \cdot B$$

Instead of transferring both A and B in FedPETuning [12], or transferring B only in FFA-LORA [54], we only transfer

Algorithm 1 Training process of CE-LoRA
___
Inputs: Client set $\{P_i\}_{i=1}^m$; Communication round T; The pre-trained model parameters W; Tri-LoRA parameters A_i, B_i, C_i for $L_i, i = 1, ..., m$; The local dataset D_i on client P_i.

1: **for** $t = 1$ to T **do**
2:     **for** each *client* $P_i \in \{P_j\}_{j=1}^m$ in parallel **do**
3:        Fine-tune local LoRA to learn A_i^t, C_i^t, B_i^t via

$$\min_{A_i, C_i, B_i} f_i(A_i, C_i, B_i | W, \bar{C}_i^{t-1}, D_i);$$

4:        Send C_i^t to the server;
5:        Keep A_i^t and B_i^t locally for next round use.
6:     **end for**
7:     *Server* receives $\{C_i^t\}_{i=1}^m$ from the clients;
8:     *Server* conducts personalized model parameter aggregation to calculate $\{\bar{C}_i^t\}_{i=1}^m$ for the clients (refer to §III-C);

9:     *Server* send $\{\bar{C}_i^t\}_{i=1}^m$ to the corresponding clients.
10: **end for**
Outputs: $\{A_i, C_i, B_i\}_{i=1}^m$.
___

C to the server for model parameter aggregation, and update A and B on the local client. Since r is much smaller than k and d, using C as the global parameter module, which is updated by aggregation on the server can significantly reduce the per-iteration communication cost between clients and server.

C. Personalized Model Parameter Aggregation

Many existing federated PFM fine-tuning methods adopt the federated averaging strategy for global model aggregation. For example, at the t-th iteration, FedPETuning [12] aggregates the local LoRA parameter matrix A and B on server side with

$$\bar{A}^t = \frac{\sum_{i=1}^m n_i A_i^t}{n}, \quad \bar{B}^t = \frac{\sum_{i=1}^m n_i B_i^t}{n},$$

where n_i and n denote the training sample number of the i-th client and the total number of training samples in all clients, respectively. However, in FL where data distribution across clients is typically non-IID, the averaging strategy tends to overlook this type of data heterogeneity.

To alleviate this issue, we design a personalized LoRA parameter aggregation strategy. Instead of learning a unique global model parameter update, we conduct client-specific model aggregation with the proposed personalized aggregation approach, that is

$$\bar{C}_i^t = \phi_i(C_0^t, C_1^t, ..., C_m^t), \forall \bar{C}_i^t \in \{\bar{C}_j^t\}_{j=1}^m \quad (2)$$

where ϕ_i refers to the personalized aggregation for deriving $\bar{C}_i^t$. It assigns higher weights to models from clients with greater similarity to client i. We formulate Eqn. 2 as

$$\bar{C}_i^t = \sum_{j=1, j \neq i}^m \frac{S_{ij}}{\sum_{j=1, j \neq i}^m S_{ij}} \cdot C_j^t \quad (3)$$

where S_{ij} represents the affinity between clients i and j.

In this work, we propose a client similarity evaluation metric, which is computed as

$$S_{ij} = S_{ij}^{\text{data}} + S_{ij}^{\text{model}} \quad (4)$$

where S_{ij}^{data} and S_{ij}^{model} denotes training dataset similarity and model similarity, respectively.

1) Training Dataset Similarity Evaluation Between Two Clients: In FL, directly accessing and comparing raw data between clients poses privacy concerns. To address this, we first learn each client's data distribution using Gaussian Mixture Models (GMM). GMM is a probabilistic generative model that represents the data distribution as a mixture of Gaussian components [59]. Suppose that there are K_i categories in the training dataset on client i, for category-k of the training dataset, we learn a GMM model with the encoder module output, that is

$$\mathcal{G}_{i,k} = \sum_{g=1}^G w_g^{(i,k)} \mathcal{N}(\mu_g^{(i,k)}, \Sigma_g^{(i,k)})$$

where G is the number of Gaussian component, the model parameters including mixture weights, mean and covariance $\{w_g^{(i,k)}, \mu_g^{(i,k)}, \Sigma_g^{(i,k)}\}_{g=1}^G$ of the Gaussian distributions are learned by EM-type algorithm. We learn the GMMs for all training data categories and obtain $\mathcal{G}_i = \{\mathcal{G}_{i,k}\}_{k=1}^{K_i}$. The client sends all parameters of $\mathcal{G}_i$ to the server to evaluate the similarity of training datasets.

Optimal transport (OT) is a mathematical framework for quantifying the distance between probability distributions by determining the most efficient transport plan that minimizes the total cost of moving the mass from one distribution to another [60]. We adopt OT to measure the training dataset similarity between $\mathcal{G}_i$ and $\mathcal{G}_j$, which is:

$$S_{ij}^{\text{data}} = \sum_{c,d} \gamma_{cd}^* GW_{cd} \quad (5)$$

where γ_{cd}^* denotes the weight of the optimal matching plan between $\mathcal{G}_i$ and $\mathcal{G}_j$, and GW_{cd} is the Wasserstein-type distance between two GMMs $\mathcal{G}_{i,c}$ and $\mathcal{G}_{j,d}$ [61].

After computing the Wasserstein distances for all category pairs between $\mathcal{G}_i$ and $\mathcal{G}_j$, we construct the category-level distance matrix $GW \in \mathbb{R}^{K_i \times K_j}$. Then we solve the OT problem to find the optimal matching between the two GMM sets via solving

$$\gamma_{cd}^* = \arg \min_{\gamma \in \Pi(\mathcal{G}_i, \mathcal{G}_j)} \sum_{c,d} \gamma_{cd} GW_{cd} \quad (6)$$

where $\Pi(\mathcal{G}_i, \mathcal{G}_j)$ denotes the set of all joint distributions with marginals $\mathcal{G}_i, \mathcal{G}_j$. It can be efficiently solved using the Sinkhorn algorithm with entropy regularization.

It is notable that the training dataset similarity evaluation is a one-shot effort. We compute the pairwise similarities of the clients at the beginning of the FL process, then use the result in the FL model learning iterations.

2) Training Model Similarity Evaluation Between Two Clients: In addition to the training dataset diversity across the clients, the model fine-tuning mechanism differences respect to hyper-parameter setting, optimizer etc. can also bring heterogeneity to the model from the involved clients.

To quantize this type of heterogeneity, we evaluate the model similarity S_{ij}^{model}, which is calculated using the Centered Kernel Alignment (CKA) method [62]. We define

$$S_{ij}^{\text{model}} = \text{CKA}(C_i^t, C_j^t). \qquad (7)$$

CKA was originally used to assess the similarity of representations between models or model layers. The computation of CKA relies on the Hilbert-Schmidt Independence Criterion (HSIC), which is used to measure the dependency between two sets of variables. Specifically, to evaluate the similarity between C_i^t and C_j^t, we first randomly generate a set with n samples, which then pass through by C_i^t and C_j^t, respectively. We use $K_i^t \in \mathbb{R}^{n \times n}$ and $K_j^t \in \mathbb{R}^{n \times n}$ to denote the linear kernel of the output. The CKA-based similarity metric is calculated via

$$\text{CKA}(C_i^t, C_j^t) = \frac{\text{HSIC}(C_i^t, C_j^t)}{\sqrt{\text{HSIC}(C_i^t, C_i^t) \cdot \text{HSIC}(C_j^t, C_j^t)}} \qquad (8)$$

where $\text{HSIC}(C_i^t, C_j^t)$ is computed by:

$$\text{HSIC}(C_i^t, C_j^t) = \text{tr}(K_i^t H K_j^t H) \qquad (9)$$

where tr denotes the matrix trace, and $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is the centering matrix.

D. Local Fine-tuning and Model Inference

For each of the clients P_i, upon receiving a global parameter update $\bar{C}_i^{t-1}$ from the server, it utilizes its local data D_i to fine-tune its LoRA parameters A_i, C_i and B_i by solving

$$\min_{A_i, C_i, B_i} f_i(A_i, C_i, B_i | W, \bar{C}_i^{t-1}, D_i).$$

The client freezes the parameters of the pre-trained model W, and initializes A, C and B with A_i^{t-1}, $\bar{C}_i^{t-1}$ and B_i^{t-1}, respectively. After reaching the convergence of model fine-tuning, the client sends C_i^t to the server for a new iteration round.

Once the federated PFM fine-tuning completes, for each layer that requires computation, the original pre-trained model weight matrix W is adjusted by the product of the trained low-rank matrices A_i, C_i, and B_i, that is

$$W_i = W + A_i \cdot C_i \cdot B_i \qquad (10)$$

where W_i is the fine-tuned weight matrix used for inference for client P_i.

IV. EXPERIMENTS

A. Experiments Setup

Models and Datasets: We evaluate four representative models spanning natural language processing and computer vision domains. For NLP tasks, we employ RoBERTa-base (125M parameters), an enhanced BERT variant optimized for language understanding, and LLaMA-7B (7B parameters), a large language model specialized for text generation and reasoning. For vision-related tasks, we utilize BLIP-2 (3B parameters), a vision-language model with two-stage image-text alignment training, and LLaVA-7B which integrates visual encoders with

Dataset	Train	Dev	Test
SST-2	66,675	674	872
MNLI	388,774	3,928	9,815
AG_NEWS	120,000	7,600	7,600
CIFAR-10	40,000	10,000	10,000
CIFAR-100	40,000	10,000	10,000
ImageNet-1K	1,231,167	50,000	50,000

TABLE I: Benchmark datasets used in the experiments.

LLaMA-7B for multimodal understanding. All models employ LoRA adaptation with default rank 8.

Our experiments use six benchmarks spanning text and image domains. For NLP datasets, SST-2 and MNLI from the GLUE benchmark are adopted for sentiment analysis and textual entailment respectively, following [63] in repurposing their original validation sets as test sets; AG_NEWS serves as a news classification benchmark. For computer vision datasets, we employ CIFAR-10, CIFAR-100 and ImageNet as benchmarks. For CIFAR-10 and CIFAR-100, which lack validation sets, we create validation sets by splitting their training data. For ImageNet, we used the official validation set for evaluation. Table I summarizes the training, validation, and testing sample sizes for these datasets.

Non-IID Data Partition on Clients: We follow the previous works [12], [64] that utilize the Dirichlet distribution $\text{Dir}(\alpha)$ to partition the dataset, in which a smaller α indicates higher data heterogeneity. In our experiments, the data is partitioned across 10 clients, and $\alpha = 0.5$ as the default setting.

Baselines: To validate the effectiveness of CE-LoRA, we compare it with six baselines: (1) LoRA training with local data; (2) FedPETuning [12], which learns vanilla LoRA in a federated approach; (3) FFA-LoRA [54], an optimized scheme within the federated learning framework that only transmits matrix B; (4) FDLoRA [56], a personalized federated method combining client-specific and global LoRA modules through adaptive fusion; (5) pFedMe-LoRA, implementing pFedMe's Moreau envelope [65] with full LoRA parameter aggregation; (6) pFedMe-FFA, a variant of pFedMe that adopts FFA-LoRA's communication mechanism to transmit only matrix B.

B. Performance Comparison

Table II and Table III respectively present the performance and communication cost comparisons of CE-LoRA against the baselines. The results indicate that CE-LoRA enhances the model's ability to adapt to local personalized data and also significantly reducing communication overhead. For example, on the CIFAR-100 dataset, CE-LoRA achieves a relative image classification accuracy improvement of over 2.29% and 1.78% for BLIP-2 and LLaVA model fine-tuning over the second-best algorithm FDLoRA. For communication cost comparison, we use FedPETuning, which transmits all LoRA parameters in each round as baseline, and evaluate the communication cost reduction. For the LLaMA model, CE-LoRA reduces communication overhead by a factor of 1024 compared to FedPETuning and by a factor of 512 compared to FFA-LoRA.

Figure 4 illustrates the performance of the best and worst clients among 10 clients in the three federated PFM fine-tuning methods. CE-LoRA consistently improves model performance

NLP Benchmarks					CV Benchmarks				
Model	Method	SST-2	MNLI	AG_NEWS	Model	Method	CIFAR-10	CIFAR-100	ImageNet
RoBERTa	LoRA-Loc	91.2	80.9	79.1	BLIP-2	LoRA-Loc	81.1	56.2	66.2
	FedPETuning	91.1	83.4	80.2		FedPETuning	82.4	59.6	66.3
	FFA-LoRA	91.7	84.2	81.9		FFA-LoRA	82.9	59.8	68.2
	pFedMe-LoRA	92.3	85.4	82.6		pFedMe-LoRA	83.5	60.9	69.3
	pFedMe-FFA	92.4	85.2	82.5		pFedMe-FFA	83.3	60.5	69.1
	FDLoRA	92.8	85.6	82.6		FDLoRA	83.7	61.1	69.4
	CE-LoRA	**93.2**	**86.1**	**83.5**		CE-LoRA	**84.4**	**62.5**	**70.1**
LLaMA	LoRA-Loc	94.3	89.1	82.3	LLaVA	LoRA-Loc	85.2	64.2	70.4
	FedPETuning	96.2	91.2	88.7		FedPETuning	85.8	66.1	71.3
	FFA-LoRA	96.2	91.7	90.0		FFA-LoRA	86.1	66.5	73.2
	pFedMe-LoRA	96.3	92.4	90.8		pFedMe-LoRA	86.8	67.6	75.4
	pFedMe-FFA	96.2	92.2	90.5		pFedMe-FFA	86.4	67.5	75.2
	FDLoRA	96.1	92.8	91.0		FDLoRA	87.0	67.6	75.9
	CE-LoRA	**96.4**	**93.1**	**91.5**		CE-LoRA	**87.2**	**68.8**	**76.6**

TABLE II: Accuracy comparison of fine-tuning methods on NLP and CV benchmarks. Left: RoBERTa and LLaMA on NLP datasets. Right: BLIP-2 and LLaVA on CV datasets.

Domain	Model	Method	Com. (Percentage)
NLP	RoBERTa	FedPETuning	2.95×10^5 (100%)
		pFedMe-LoRA	2.95×10^5 (100%)
		FDLoRA	2.95×10^5 (100%)
		pFedMe-FFA	1.47×10^5 (50%)
		FFA-LoRA	1.47×10^5 (50%)
		CE-LoRA	$\mathbf{7.68 \times 10^2}$ (0.26%)
	LLaMA	FedPETuning	4.19×10^6 (100%)
		pFedMe-LoRA	4.19×10^6 (100%)
		FDLoRA	4.19×10^6 (100%)
		pFedMe-FFA	2.10×10^6 (50%)
		FFA-LoRA	2.10×10^6 (50%)
		CE-LoRA	$\mathbf{4.10 \times 10^3}$ (0.10%)
CV	BLIP-2	FedPETuning	2.36×10^6 (100%)
		pFedMe-LoRA	2.36×10^6 (100%)
		FDLoRA	2.36×10^6 (100%)
		pFedMe-FFA	1.18×10^6 (50%)
		FFA-LoRA	1.18×10^6 (50%)
		CE-LoRA	$\mathbf{6.14 \times 10^3}$ (0.26%)
	LLaVA	FedPETuning	4.98×10^6 (100%)
		pFedMe-LoRA	4.98×10^6 (100%)
		FDLoRA	4.98×10^6 (100%)
		pFedMe-FFA	2.49×10^6 (50%)
		FFA-LoRA	2.49×10^6 (50%)
		CE-LoRA	$\mathbf{7.17 \times 10^3}$ (0.14%)

TABLE III: Communication cost comparison (Com. indicates parameters transmitted per iteration, percentage shows compression ratio relative to FedPETuning)

for the considered models and datasets and significantly outperforms the other methods in the worst-performing client. The results indicate that CE-LoRA can effectively improve the model performance for clients with insufficient training data, which depend on FL to help improve the model performance.

C. Privacy Protection Experiment

For privacy protection evaluation, following [12], we employ the DLG method [66], an attack technique that reconstructs data from gradients. We measure precision (the average percentage of correctly recovered words in the target text), recall (the
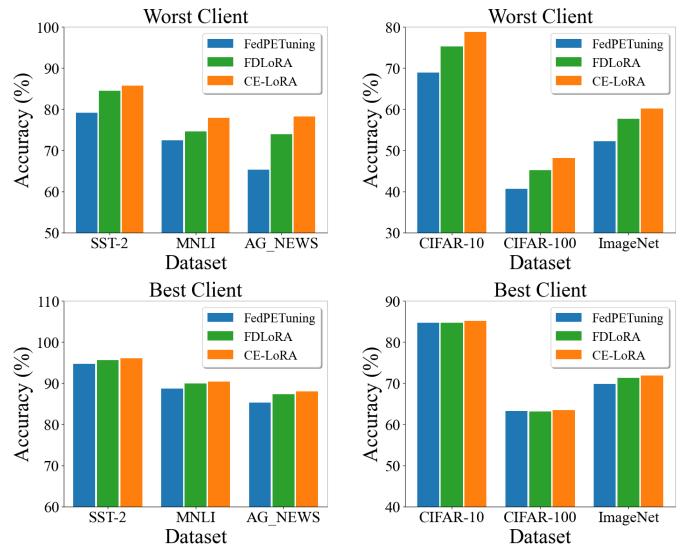

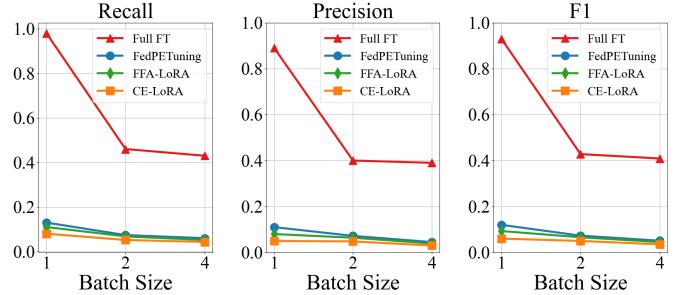Fig. 4: Performance comparison of worst-performing client and best-performing client.

Fig. 5: The comparison of the data reconstruction attack on the dataset using full PFM fine-tuning, FedPETuning, FFA-LoRA and CE-LoRA.

average percentage of words in the target text that were correctly predicted) and the F1 score (the harmonic mean of precision and recall). We randomly select 128 samples from MNLI as the attack dataset.

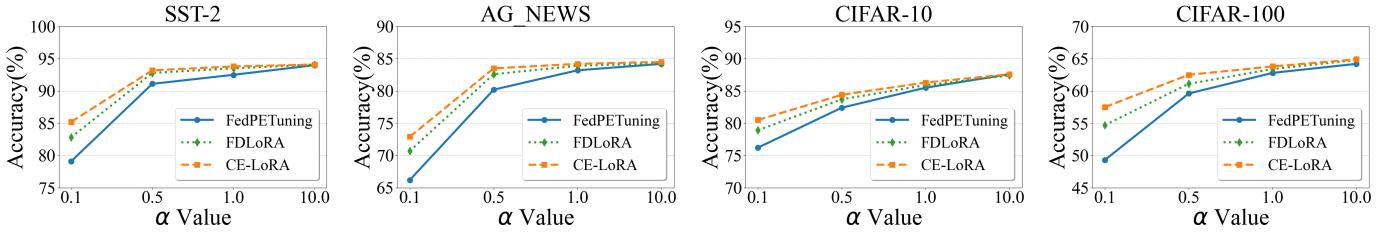Figure 5 shows the DLG attack results of four representative

Fig. 6: Performance comparison of FedPETuning, FDLoRA and CE-LoRA under different data heterogeneity degree parameterized by varying α values.

Method	SST-2	MNLI	AG_NEWS
LoRA + FedAvg	91.1	83.4	80.2
Tri-LoRA + FedAvg	91.2	83.3	81.5
Tri-LoRA + S^{data}	92.6	83.7	81.9
Tri-LoRA + S^{data} + S^{model}	**93.2**	**86.1**	**83.5**

TABLE IV: Ablation study experiment results on NLP datasets using RoBERTa. "Tri-LoRA" represents the proposed triple LoRA factorization discussed in §III-B. "S^{data}" and "S^{model}" represent dataset similarity and model similarity for personalized model parameter aggregation, detailed in §III-C.

Method	CIFAR-10	CIFAR-100	ImageNet
LoRA + FedAvg	85.8	66.1	71.3
Tri-LoRA + FedAvg	85.9	66.4	71.6
Tri-LoRA + S^{data}	87.1	66.9	75.8
Tri-LoRA + S^{data} + S^{model}	**87.2**	**68.8**	**76.6**

TABLE V: Ablation study experiment results on CV datasets using LLaVA. Method notations are consistent with Table IV.

methods with different communication transmission matrices. The results indicate that CE-LoRA can defend against data reconstruction attacks more effectively than other fine-tuning methods, and its privacy protection performance is relatively unaffected by batch sizes. This is because CE-LoRA only transmits $r \times r$ parameters of the LoRA model per-iteration, and it is more difficult for attackers to reconstruct data from such a limited number of model parameters.

D. Ablation Study

In this section, we conducted three ablation experiments on the RoBERTa and LLaVA models to validate the effectiveness of our proposed matrix decomposition and personalized model aggregation method. The experimental results are shown in Table IV and Table V. We first compare our triple LoRA factorization method with vanilla LoRA while using federated average as the model aggregation strategy. The result demonstrates that the proposed LoRA factorization can maintain performance with significantly reduced communication overhead. Then we test applying personalized model aggregation over using federated average. Compared to traditional federated average, the proposed personalized global aggregation effectively improves the performance of the model in heterogeneous environments.

E. Performance Comparison under Varying Data Skew

We evaluate FedPETuning, FDLoRA, and CE-LoRA under varying data heterogeneity levels, fine-tuning RoBERTa on
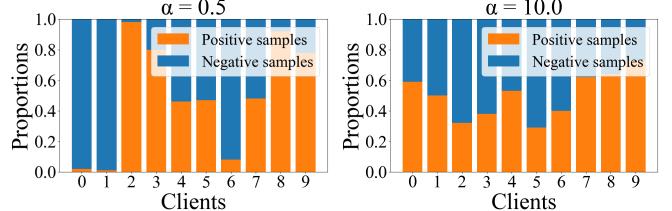

Fig. 7: The training set label distribution of the SST-2 dataset under different Dirichlet parameter α for the 10 clients.

SST-2 and AG_NEWS, and BLIP-2 on CIFAR-10 and CIFAR-100. In this experiment, four Dirichlet distributions are tested by setting α as {0.1, 0.5, 1.0, 10.0}. As shown in Figure 7, the smaller the α, the more heterogeneous data distribution between clients. When α is 10.0, the data across most clients is nearly uniformly distributed.

As shown in Figure 6, CE-LoRA, FedPETuning, and FDLoRA are found to have a decrease in performance as the value of α gets decreases. However, CE-LoRA is clearly less affected by data heterogeneity. This is because CE-LoRA employs a personalized model aggregation scheme on LoRA component C, together with LoRA local fine-tuning for the component A and B, which leads to better model generalization to local data.

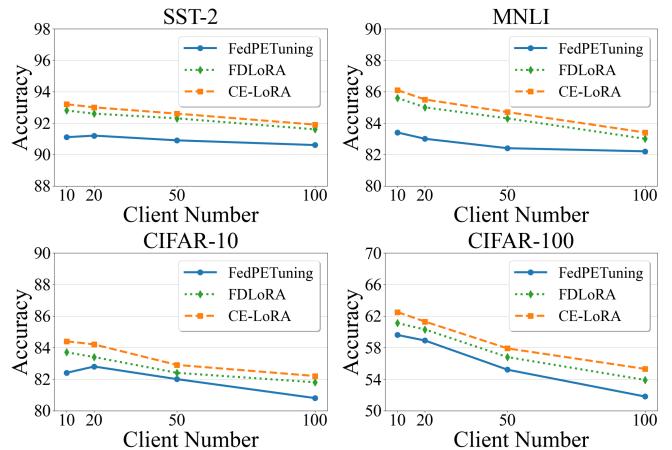### F. Performance Comparison under Varying Client Numbers

Fig. 8: Performance comparison of FedPETuning, FDLoRA, and CE-LoRA under varying clients numbers.

We compare the performance of FedPETuning, FDLoRA, and CE-LoRA by partitioning the benchmark datasets into a
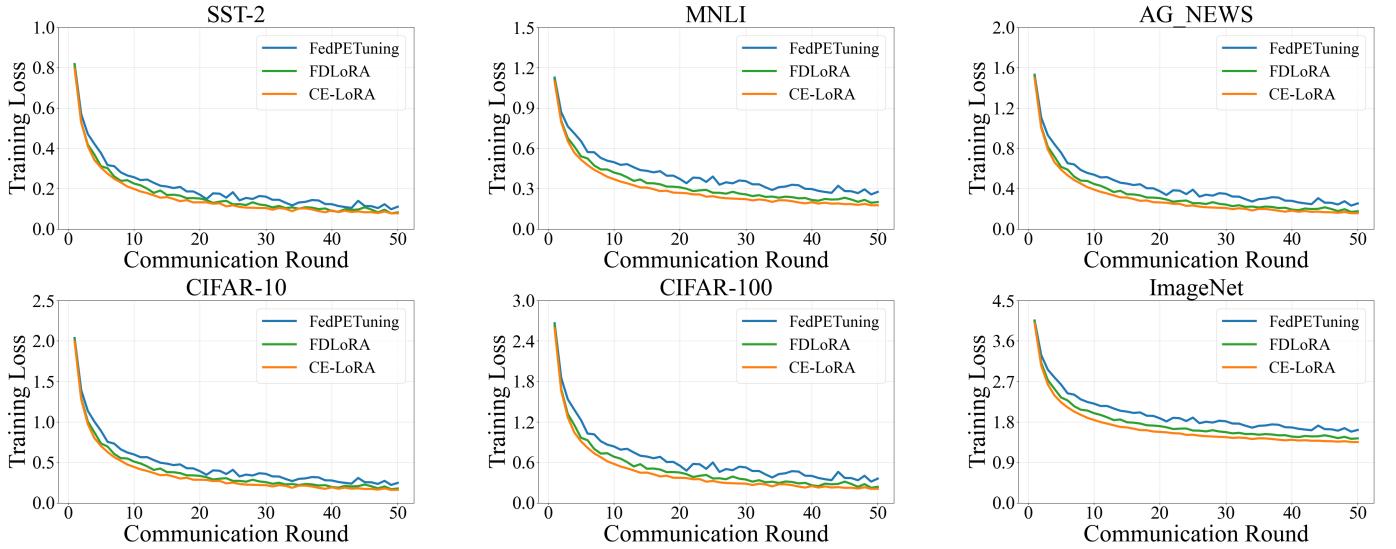
Fig. 9: RoBERTa and BLIP-2 fine-tuning convergence comparison of FedPETuning, FDLoRA, and CE-LoRA on the considered benchmark datasets.

varying number of clients. We fine-tune RoBERTa on SST-2 and MNLI and BLIP-2 on Cifar10 and Cifar100, respectively. The number of clients is set to 10, 20, 50, and 100. As shown in Figure 8, CE-LoRA consistently outperforms both FedPETuning and FDLoRA as the number of clients increases.
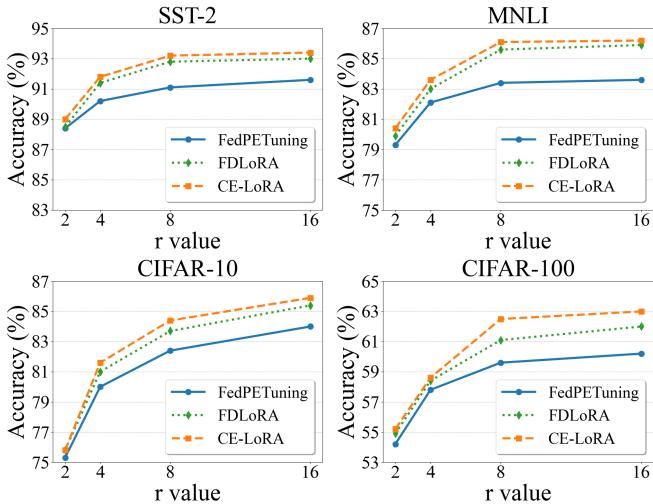
G. Hyperparameter Analysis

Fig. 10: Performance comparison of FedPETuning, FDLoRA, and CE-LoRA under different ranks.

In this part, we set up an experiment to observe how performance and communication costs change as the rank r of the low-rank matrices in LoRA changes. We fine-tune RoBERTa on SST-2 and MNLI and BLIP-2 on Cifar10 and Cifar100, respectively. The communication cost increases as $O(r^2)$ with increasing rank r. As shown in Figure 10, CE-LoRA achieves a 3.1% performance gain when increasing r from 2 to 4, while extending r from 8 to 16 yields merely a 0.2% improvement despite inducing a $16\times$ larger communication

overhead increment compared to the former scaling. There is a trade-off between limited resources and performance, making $r = 4$ or $r = 8$ often a better choice.

H. FL Convergence Analysis

Figure 9 shows the convergence of the FL fine-tuning with respect to the communication round for RoBERTa and BLIP-2 on the benchmark datasets. The results show that CE-LoRA achieves faster convergence and less training loss variation than FedPETuning and FDLoRA. These advantages first stem from CE-LoRA's personalized aggregation strategy, which is based on inter-client data and model similarity, effectively mitigating parameter conflicts of model from different clients. Secondly, the fine-tuning of LoRA component C only in federated approach reduces the parameter number to be learned by FL, which also facilitates faster convergence.

I. Computational Overhead of Personalized Aggregation

	RoBERTa		BLIP-2	
CPU	SST-2	MNLI	CIFAR-10	CIFAR-100
1	14.85	15.04	20.03	20.14
5	4.05	4.11	5.49	5.58
10	2.49	2.53	3.38	3.41
20	1.48	1.51	2.11	2.12

TABLE VI: Computational Overhead of CE-LoRA Aggregation with 100 Clients (seconds).

The computational complexity of the model similarity evaluation presented in is $O(m^2)$ where m is the number of clients. However, since the transmitted LoRA component C is compact, the similarity computation is lightweight. The overall computation can be further speed up by parallel computing. As shown in Table VI, we evaluate the time cost of pairwise model similarity computation across 100 clients under varying levels of CPU parallelism with 1, 5, 10 and 20 CPUs, respectively.

For even larger number of clients, using hierarchical structure such as the client-edge-cloud aggregation hierarchy [67] and client sampling [68] can handle the scalability problem.

V. CONCLUSION

This paper introduces the CE-LoRA method, which leverages triple LoRA factorization and personalized global model aggregation to address the challenges of high communication overhead and suboptimal performance on heterogeneous data in federated learning environments. Experimental results demonstrate that CE-LoRA not only reduces communication costs but also significantly enhances model performance on non-IID data, particularly in environments with highly heterogeneous data distributions. Moreover, it excels in protecting data privacy, effectively resisting gradient-based data reconstruction attacks. Our work primarily addresses data heterogeneity, with future research extending CE-LoRA to task and model heterogeneity to broaden its applicability in federated learning.

REFERENCES

[1] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *NeurIPS*, 2020.

[3] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao, "Llava-med: Training a large language-and-vision assistant for biomedicine in one day," *arXiv preprint arXiv:2306.00890*, 2023.

[4] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *NeurIPS*, 2023.

[5] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *NeurIPS*, 2022.

[6] X. Lin, W. Wang, Y. Li, S. Yang, F. Feng, Y. Wei, and T.-S. Chua, "Data-efficient fine-tuning for llm-based recommendation," in *SIGIR*, 2024.

[7] X. Liu, K. Ji, Y. Fu, W. L. Tam, Z. Du, Z. Yang, and J. Tang, "P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks," in *ACL*, 2022.

[8] J.-G. Zhang, T. Bui, S. Yoon, X. Chen, Z. Liu, C. Xia, Q. H. Tran, W. Chang, and P. Yu, "Few-shot intent detection via contrastive pre-training and fine-tuning," in *EMNLP*, 2021.

[9] V. Thangarasa, A. Gupta, W. Marshall, T. Li, K. Leong, D. DeCoste, S. Lie, and S. Saxena, "Spdf: Sparse pre-training and dense fine-tuning for large language models," in *UAI*, 2023.

[10] A. Köpf, Y. Kilcher, D. von Rütte, S. Anagnostidis, Z. R. Tam, K. Stevens, A. Barhoum, D. Nguyen, O. Stanley, R. Nagyfi *et al.*, "Openassistant conversations-democratizing large language model alignment," *NeurIPS*, 2024.

[11] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *ICLR*, 2022.

[12] Z. Zhang, Y. Yang, Y. Dai, Q. Wang, Y. Yu, L. Qu, and Z. Xu, "Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models," in *ACL*, 2023.

[13] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[14] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[15] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, pp. 1–45, 2024.

[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *JMLR*, vol. 21, no. 140, pp. 1–67, 2020.

[17] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[18] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI*, 2019.

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

[21] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *JMLR*, vol. 23, no. 120, pp. 1–39, 2022.

[22] A. Clark, D. de Las Casas, A. Guy, A. Mensch, M. Paganini, J. Hoffmann, B. Damoc, B. Hechtman, T. Cai, S. Borgeaud *et al.*, "Unified scaling laws for routed language models," in *ICML*, 2022.

[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[24] Z. Liang, Y. Xu, Y. Hong, P. Shang, Q. Wang, Q. Fu, and K. Liu, "A survey of multimodal large language models," in *ICCAC*, 2024.

[25] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao, "Llava-med: Training a large language-and-vision assistant for biomedicine in one day," *NeurIPS*, 2023.

[26] J. Tang, Y. Yang, W. Wei, L. Shi, L. Su, S. Cheng, D. Yin, and C. Huang, "Graphgpt: Graph instruction tuning for large language models," in *SIGIR*, 2024.

[27] D. J. Kopiczko, T. Blankevoort, and Y. M. Asano, "Vera: Vector-based random matrix adaptation," in *ICLR*, 2024.

[28] Q. Zhang, M. Chen, A. Bukharin, N. Karampatziakis, P. He, Y. Cheng, W. Chen, and T. Zhao, "Adalora: Adaptive budget allocation for parameter-efficient fine-tuning," in *ICLR*, 2023.

[29] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient fine-tuning of quantized llms," *NeurIPS*, 2024.

[30] N. Ding, X. Lv, Q. Wang, Y. Chen, B. Zhou, Z. Liu, and M. Sun, "Sparse low-rank adaptation of pre-trained language models," *arXiv preprint arXiv:2311.11696*, 2023.

[31] X. Wu, S. Huang, and F. Wei, "Mixture of lora experts," in *ICLR*, 2024.

[32] W. Feng, C. Hao, Y. Zhang, Y. Han, and H. Wang, "Mixture-of-loras: An efficient multitask tuning method for large language models," in *LREC-COLING*, 2024.

[33] G. Long, "The rise of federated intelligence: From federated foundation models toward collective intelligence," in *IJCAI*, 2024.

[34] H. Chen, Y. Zhang, D. Krompass, J. Gu, and V. Tresp, "Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning," in *AAAI*, 2024.

[35] Y. Bao, "Prompt tuning empowering downstream tasks in multimodal federated learning," in *BESC*. IEEE, 2024.

[36] F. Wu, Z. Li, Y. Li, B. Ding, and J. Gao, "Fedbiot: Llm local fine-tuning in federated learning without full model," in *KDD*, 2024.

[37] Z. Guo, Y. Zhang, Z. Zhang, Z. Xu, and I. King, "Fedlfc: Towards efficient federated multilingual modeling with lora-based language family clustering," in *NAACL*, 2024.

[38] ——, "Fedhlt: Efficient federated low-rank adaptation with hierarchical language tree for multilingual modeling," in *WWW*, 2024.

[39] Y. Liu, X. Bi, L. Li, S. Chen, W. Yang, and X. Sun, "Communication efficient federated learning for multilingual neural machine translation with adapter," in *ACL*, 2023.

[40] T. Che, J. Liu, Y. Zhou, J. Ren, J. Zhou, V. S. Sheng, H. Dai, and D. Dou, "Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization," in *EMNLP*, 2023.

[41] C. Dong, Y. Xie, B. Ding, Y. Shen, and Y. Li, "Tunable soft prompts are messengers in federated learning," in *EMNLP*, 2023.

[42] R. Ye, W. Wang, J. Chai, D. Li, Z. Li, Y. Xu, Y. Du, Y. Wang, and S. Chen, "Openfedllm: Training large language models on decentralized private data via federated learning," in *KDD*, 2024.

[43] W. Kuang, B. Qian, Z. Li, D. Chen, D. Gao, X. Pan, Y. Xie, Y. Li, B. Ding, and J. Zhou, "Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning," in *KDD*, 2024.

[44] M. Vu, T. Nguyen, M. T. Thai *et al.*, "Analysis of privacy leakage in federated large language models," in *AISTATS*, 2024.

[45] S. Han, B. Buyukates, Z. Hu, H. Jin, W. Jin, L. Sun, X. Wang, W. Wu, C. Xie, Y. Yao *et al.*, "Fedsecurity: A benchmark for attacks and defenses in federated learning and federated llms," in *KDD*, 2024.

[46] Z. Qin, D. Chen, B. Qian, B. Ding, Y. Li, and S. Deng, "Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes," in *ICML*, 2024.

[47] Z. Li, Z. Hou, H. Liu *et al.*, "Federated learning in large model era: Vision-language model for smart city safety operation management," in *WWW Companion*. ACM, 2024.

[48] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE TNNLS*, vol. 34, no. 12, pp. 9587–9603, 2022.

[49] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *MLSys*, 2020.

[50] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *CVPR*, 2021.

[51] X. Ma, J. Zhang, S. Guo, and W. Xu, "Layer-wised model aggregation for personalized federated learning," in *CVPR*, 2022.

[52] Y. Shen, Y. Zhou, and L. Yu, "Cd2-pfed: Cyclic distillation-guided channel decoupling for model personalization in federated learning," in *CVPR*, 2022.

[53] J. Wang, X. Yang, S. Cui, L. Che, L. Lyu, D. D. Xu, and F. Ma, "Towards personalized federated learning via heterogeneous model reassembly," *NeurIPS*, 2024.

[54] Y. Sun, Z. Li, Y. Li, and B. Ding, "Improving lora in privacy-preserving federated learning," *ICLR*, 2024.

[55] K. Kuo, A. Raje, K. Rajesh, and V. Smith, "Sparsity for communication-efficient lora," in *PMLR*, 2024.

[56] J. Qi, Z. Luan, S. Huang, C. Fung, H. Yang, and D. Qian, "Fdlora: Personalized federated learning of large language model via dual lora tuning," *arXiv preprint arXiv:2406.07925*, 2024.

[57] Y. Yang, G. Long, T. Shen, J. Jiang, and M. Blumenstein, "Dual-personalizing adapter for federated foundation models," *arXiv preprint arXiv:2403.19211*, 2024.

[58] Z. Wang, Z. Shen, Y. He, G. Sun, H. Wang, L. Lyu, and A. Li, "Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations," *NeurIPS*, 2024.

[59] D. A. Reynolds *et al.*, "Gaussian mixture models." *Encyclopedia of biometrics*, vol. 741, no. 659-663, p. 3, 2009.

[60] G. Peyré, M. Cuturi *et al.*, "Computational optimal transport: With applications to data science," *Found. Trends Mach. Learn.*, 2019.

[61] J. Delon and A. Desolneux, "A wasserstein-type distance in the space of gaussian mixture models," *SIAM J. Imaging Sci.*, vol. 13, no. 2, pp. 936–970, 2020.

[62] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *ICML*, 2019.

[63] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.

[64] D. Wang, L. Shen, Y. Luo, H. Hu, K. Su, Y. Wen, and D. Tao, "Fedabc: Targeting fair competition in personalized federated learning," in *AAAI*, 2023.

[65] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *NeurIPS*, 2020.

[66] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *NeurIPS*, 2019.

[67] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Hierarchical federated learning with quantization: Convergence analysis and system design," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 2–18, 2022.

[68] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *Trans. Mach. Learn. Res.*, 2022.

Bo Liu received the Ph.D. degree in computer science from Rutgers, The State University of New Jersey, in 2018. He currently is a faculty member at Hefei University of Technology, Hefei, China. Prior to this role, He held the positions of Staff Data Scientist and Senior Research Scientist at Walmart Global Tech and JD.com Silicon Valley Research Center, respectively. His areas of expertise and research interests encompass machine learning, computer vision, and data science.

Sheng Huang (Member, IEEE) received the B.Eng. and Ph.D. degrees from Chongqing University, Chongqing, China, in 2010 and 2015, respectively. He was a visiting Ph.D. Student with the Department of Computer Science, Rutgers University, New Brunswick, NJ, USA, from 2012 to 2014. He is currently a full Professor with the School of Big Data and Software Engineering, Chongqing University, China. His research interests include pattern recognition, computer vision and medical image analysis.

Zheng Zhang Graduated from Tsinghua University, he has over 7 years of professional experience, consistently working in start-up companies. They possess extensive research and industrial expertise in deep learning, natural language processing (NLP) technologies, and large-scale models. he has led the development of initiatives such as the Atom Large Language Model, demonstrating profound contributions to both academic research and real-world applications in these fields.

XiaoTong Yuan (Member, IEEE) received the B.A. in computer science from Nanjing University of Posts and Telecommunications in 2002, the M.E. in electrical engineering from Shanghai Jiao-Tong University in 2005, and the Ph.D. in pattern recognition from the Chinese Academy of Sciences in 2009. He then worked as a Postdoctoral Research Associate at the National University of Singapore, Rutgers University, and Cornell University. In 2013, he joined Nanjing University of Information Science and Technology, where he is currently a Professor of computer science. His main research interests include machine learning, optimization, and computer vision.

Richang Hong is a professor and PhD supervisor at Hefei University of Technology. He currently serves as the Dean of the School of Computer and Information (School of Artificial Intelligence) and the School of Software at Hefei University of Technology. He is also the Deputy Director of the Data Space Research Institute at the Hefei Comprehensive National Science Center and the President of the Anhui Artificial Intelligence Society. His research focuses on artificial intelligence-related fields, with over 300 high-level papers published and more than 20,000 citations. He serves as a Steering Committee member of the International Conference on Multimedia Modeling and as an editorial board member of six international journals, including IEEE TBD, IEEE TMM, and ACM TOMM. He has led various national projects, including the Ministry of Science and Technology's 863 Program, the Ministry's Key Research and Development Program, and projects funded by the National Natural Science Foundation of China, such as the Excellent Young Scientists Fund and key foundation projects. His research achievements have been recognized with the National Natural Science Award (Second Prize, 2015), Anhui Provincial Natural Science Award (First Prize, 2017), and Anhui Provincial Science and Technology Progress Award (First Prize, 2020). In teaching, he received the Anhui Provincial Teaching Achievement Award (First Prize, 2022) and the National Teaching Achievement Award (First Prize for Postgraduate Education, 2023). He was awarded the Anhui Province Youth May Fourth Medal in 2019 and was selected for the national leading talent program.

Yongle Li is currently a PhD candidate at the School of Computer and Information, Hefei University of Technology, China, under the supervision of Prof. Richang Hong (since 2023). His research interests include fine-tuning pre-trained founda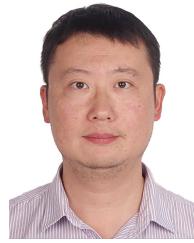tion models and federated learning.