Collaborative LLM Fine-Tuning over Mobile Networks via Sparse-and-Orthogonal LoRA

Nuocheng Yang*, Sihua Wang*, Ouwen Huan*, Mingzhe Chen‡, Changchuan Yin*
*Beijing Laboratory of Advanced Information Network, Beijing University of Posts and Telecommunications, Beijing, China, Emails: {yangnuocheng, sihuawang, ouwenh, ccyin}@bupt.edu.cn.
‡Department of Electrical and Computer Engineering and the Institute for Data Science and Computing, University of Miami, Coral Gables, FL USA, Email: mingzhe.chen@miami.edu.

***Abstract*—Distributed federated learning (DFL) enables mobile devices to fine-tune a large language model (LLM) collaboratively by directly sharing local updated parameters on a device-to-device (D2D) network through wireless connections. However, the local fine-tuning and updated parameters aggregation impose a significant burden on devices with limited computational and communication resources. Furthermore, aggregating parameters from heterogeneous datasets can lead to catastrophic knowledge interference, thereby degrading the overall performance of LLM. To address these issues, we formulate an optimization problem whose goal is to maximize LLM's overall performance under resource constraint. To address this problem, we employ a sparse-LoRA method that reduces both computational and communication overhead by decomposing parameters into low-rank and sparse activated matrices. To further enhance the DFL performance, we impose a parameter-orthogonalization constraint on the LoRA matrix on each device to mitigate knowledge interference. Finally, we analyze the relationship between D2D topology and DFL performance, and propose a cluster-based topology design for efficient parameter aggregation. Simulation results demonstrate that the proposed approach effectively reduces communication resources consumption by 55% and enhances DFL performance by 22% compared to baselines.**

***Keywords*—Distributed federated learning, large language model (LLM), Interference-free LoRA, Sparse activation LoRA.**

I. Introduction

Large language models (LLMs) have shown their comprehension and reasoning capabilities on general tasks, superior to traditional artificial intelligence (AI) models that are tailored for a single task [1]–[6]. Although LLMs possess substantial general knowledge, continuous fine-tuning based on the devices' dataset remains essential for adapting them to specific downstream tasks. However, standard fine-tuning methods [3] require a centralized server to collect raw data from massive wireless connected devices, which introduces significant dataset transmission overhead and privacy concerns. Federated learning (FL) addresses this issue by enabling devices to collaboratively train an LLM through exchanging local fine-tuned parameter updates with a centralized server for knowledge aggregation through wireless connection, rather than massive data transmission. Since FL still relies on a centralized server that needs to communicate with all devices simultaneously, which can introduce a communication bottleneck on the server and a single-point failure risk [7]. Distributed FL (DFL) is a potential solution to address these issues, which enables devices to exchange local fine-tuned parameter updates with a subset of neighbors without relying on a centralized server. However, fine-tuning LLM in the DFL framework poses two key challenges [8]. First, fine-tuning LLMs in DFL demands substantial computational and communication resources due to their dense parameters, posing a significant burden for constrained and heterogeneous resources on devices. Secondly, the aggregation of knowledge updates from devices with heterogeneous datasets can lead to knowledge interference, ultimately compromising the LLM's fine-tuning performance and stability.

Recently, a number of works such as [9]–[11] have studied a range of parameter-efficient fine-tuning techniques to reduce the computational and communication overhead in LLM fine-tuning. The authors in [9] first developed a low-rank adaptation (LoRA) fine-tuning method, which decomposes the LLMs' dense parameters into the product of two low-rank matrices, thus reducing the number of parameters that need to be updated and shared. The authors in [10] optimized the rank of the decomposed matrices based on devices' heterogeneous transmission resources in FL framework. Furthermore, the authors in [11] developed a parameter-sparse activation LoRA method that only updates and shares a subset of parameters in LoRA, thus reducing both the transmission and computing overhead. Despite parameter-efficient methods effectively reducing the computational and transmission overhead, these works [9]–[11] remain susceptible to knowledge interference arising from aggregation parameters, which exhibit divergent update directions on heterogeneous datasets,

Some related works [12], [13] studied knowledge interference-free methods in the multi-task fine-tuning scenario. The authors in [12] attributed the knowledge interference among tasks to the overlap of their gradient update subspaces. Then, they proposed an orthogonal gradient descent method to force the subspaces of updates to be orthogonal with each other, thus reducing knowledge conflicts. Inspired by [12], the authors in [13] proposed a parameter-orthogonal-based LLM fine-tuning method in FL, where a centralized server collects and redesigns updating subspaces for each device. However, these works [12], [13] can only be employed in an FL scenario, as a centralized server is required to maintain the orthogonality between devices' update

This work is supported in part by the National Natural Science Foundation of China under Grant 62471056 and 62501072, in part by BUPT2025KYQD03, and in part by CSC scholarship 202506470060.

subspaces. On the contrary, devices in DFL cannot collect global parameters to ensure orthogonality due to their limited communication resources. Thus, the challenge of parameter subspace orthogonalization in DFL remains unresolved. The authors in [14] proposed an adaptive LoRA method in the DFL framework, considering data heterogeneity. However, the authors in [14] employed a fixed device connection topology, failing to account for its fundamental influences on DFL performance and transmission resource consumption.

In this paper, we propose a novel sparse-LoRA and parameter orthogonal LoRA, alongside a cluster-based device connection topology design strategy. The joint design of these components effectively reduces both computational and communication overhead as well as mitigates knowledge interference under heterogeneous resource constraints in DFL. Our key contributions are as follows:

- We formulated an optimization problem whose goal is to maximize LLM fine-tuning performance on all tasks while satisfying the resource and delay requirements in the DFL framework. We introduce a sparse-LoRA method to reduce the computational and communication resources consumption during DFL fine-tuning. Subsequently, we propose a parameter orthogonalization method to mitigate knowledge interference during DFL parameter aggregation.
- To further enhance DFL performance, we analyze how device connection topology affects the gap between the optimal and aggregated models by jointly considering the proposed sparse-and-orthogonal LoRA fine-tuning method. Based on the analysis, we proposed a cluster-based device connection topology design method.

An extensive experiment shows that the proposed method can effectively reduce communication resources consumption by up to 55% as well as improve overall LLM performance by up to 22% through device connection topology design compared with LoRA method.

II. System Model and Problem Formulation

Consider a distributed wireless network consisting of a set $\mathcal{M}$ of M devices that collaboratively train an LLM that can adapt to multiple downstream tasks. We assume that each device i in $\mathcal{M}$ has a local dataset $\mathcal{D}_i$ for a unique downstream task K_i, containing N_i data samples with $N = \sum_{i=1}^{M} N_i$ being the total number of data samples across devices. Each data sample n consists of an input feature vector $\boldsymbol{x}_{i,n} \in \mathbb{R}^{N_{\mathrm{I}} \times 1}$ and a corresponding label vector $\boldsymbol{y}_{i,n} \in \mathbb{R}^{N_{\mathrm{O}} \times 1}$. Each device i aims to collaborative solve the following loss function $F(\cdot)$ minimization problem across datasets:

$$\min_{\boldsymbol{w}_{1,t},\ldots,\boldsymbol{w}_{M,t}} \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{M} F\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \mathcal{D}_j\right), \tag{1}$$

where $\boldsymbol{W}_0 \in \mathbb{R}^{d \times k}$ is the fixed pre-trained LLM parameters and $\boldsymbol{w}_{i,t} \in \mathbb{R}^{d \times k}$ is the trainable adapter parameters of device i at iteration t. In particular, loss function $F\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \mathcal{D}_i\right)$ is given by

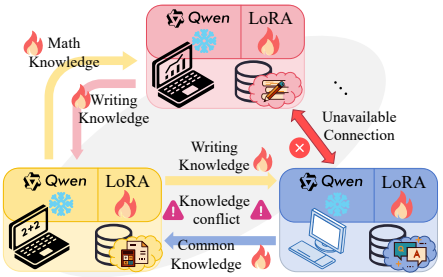

Fig. 1. Illustration of the considered Multi-task DFL framework.

$$F\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \mathcal{D}_i\right) = \frac{1}{N_i} \sum_{n=1}^{N_i} f\left(\phi\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \boldsymbol{x}_{i,n}\right), \boldsymbol{y}_{i,n}\right), \tag{2}$$

where $\phi\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \boldsymbol{x}_{i,n}\right)$ denotes the LLM's output based on $\boldsymbol{W}_0$, $\boldsymbol{w}_{i,t}$, and $\boldsymbol{x}_{i,n}$. $f\left(\phi\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \boldsymbol{x}_{i,n}\right), \boldsymbol{y}_{i,n}\right)$ is the loss function that measures the difference between the output $\phi\left(\boldsymbol{W}_0, \boldsymbol{w}_m, \boldsymbol{x}_{i,n}\right)$ and label $\boldsymbol{y}_{i,n}$.

A. DFL Local Fine-tuning Process

To reduce the computation and transmission overhead caused by full parameter fine-tuning, LoRA is utilized to decompose $\boldsymbol{w}_{i,t} \in \mathbb{R}^{d \times k}$ into two low-rank matrices $\boldsymbol{A}_{i,t} \in \mathbb{R}^{d \times r}$ and $\boldsymbol{B}_{i,t} \in \mathbb{R}^{r \times k}$ with fewer parameters, which is given by

$$\boldsymbol{w}_{i,t} = \boldsymbol{A}_{i,t} \boldsymbol{B}_{i,t}, \tag{3}$$

where $\boldsymbol{A}_{i,t}$ is called projection matrix and $\boldsymbol{B}_{i,t}$ is called expansion matrix, respectively. Both matrices share a low rank $r \ll \min(d, k)$, where d and k are their dimensions. Generally, in each iteration t, each device i performs a local update based on the local available dataset $\mathcal{D}_i$ as follows

$$\boldsymbol{A}'_{i,t+1} = \boldsymbol{A}_{i,t} - \eta \nabla_{\boldsymbol{A}_{i,t}} F\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \mathcal{D}_i\right), \tag{4}$$

$$\boldsymbol{B}'_{i,t+1} = \boldsymbol{B}_{i,t} - \eta \nabla_{\boldsymbol{B}_{i,t}} F\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \mathcal{D}_i\right), \tag{5}$$

where η is the learning rate, $\nabla_{\boldsymbol{A}_{i,t}} F\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \mathcal{D}_i\right)$ and $\nabla_{\boldsymbol{B}_{i,t}} F\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \mathcal{D}_i\right)$ are the cumulative gradient.

To further reduce the number of parameters needed to be updated in $\boldsymbol{A}_{i,t}$ and $\boldsymbol{B}_{i,t}$, we employ a parameter sparsification method, which only updates a subset of the most important parameters under resource constraints. The sparsification mask for $\boldsymbol{A}_{i,t}$ and $\boldsymbol{B}_{i,t}$ are $\boldsymbol{M}_{i,t}^{A} \in \{0,1\}^{d \times r}$ and $\boldsymbol{M}_{i,t}^{B} \in \{0,1\}^{r \times k}$, respectively. $\left[\boldsymbol{M}_{i,t}^{A}\right]_{x,y} = 1$ implies that the $\{x, y\}$-th parameter in $\boldsymbol{A}_{i,t}$ is trainable, and $\left[\boldsymbol{M}_{i,t}^{A}\right]_{x,y} = 0$, otherwise. Thus, the updated adapter of device i can be given by

$$\boldsymbol{w}_{i,t} = \left(\boldsymbol{A}'_{i,t+1} \odot \boldsymbol{M}_{i,t}^{A}\right)\left(\boldsymbol{B}'_{i,t+1} \odot \boldsymbol{M}_{i,t}^{B}\right), \tag{6}$$

where $\odot$ is the element-wise multiplication function. Then, we can define the sparsity rate of mask $\boldsymbol{M}_{i,t}^{A}$ as $s_{\boldsymbol{M}_{i,t}^{A},t} = \frac{||\boldsymbol{M}_{i,t}^{A}||}{d \times r}$, which refers to the probability that any single parameter position is active for fine-tuning.

The computing overhead $c_{i,t}$ for device i in local fine-tuning process is determined by the number of parameters that need to be updated $Q_{i,t}$, the size of the dataset $|\mathcal{D}_i|$, and the fine-tuning batch size B, which can be given by:

$$c_{i,t} = \frac{\alpha Q_{i,t} |\mathcal{D}_i|}{B}, \tag{7}$$

where α is the computing correlation coefficient related to the device's computing power.

B. DFL Model Aggregation Process

After local fine-tuning, each device i will exchange the updated LoRA parameters with a subset of its neighbors for aggregation, which is given by [15]:

$$\boldsymbol{A}_{i,t+1}=\boldsymbol{A}'_{i,t}-\sum_{j\in\mathcal{M}} \frac{u_{j,i,t}}{\max\{||\boldsymbol{u}_{i,t}||, ||\boldsymbol{u}_{j,t}||\}}(\boldsymbol{A}'_{j,t}-\boldsymbol{A}'_{i,t}), \tag{8}$$

$$\boldsymbol{B}_{i,t+1}=\boldsymbol{B}'_{i,t}-\sum_{j\in\mathcal{M}} \frac{u_{j,i,t}}{\max\{||\boldsymbol{u}_{i,t}||, ||\boldsymbol{u}_{j,t}||\}}(\boldsymbol{B}'_{j,t}-\boldsymbol{B}'_{i,t}), \tag{9}$$

where $\boldsymbol{u}_{i,t} = [u_{i,1,t}, \cdots, u_{i,M,t}]$ is a model transmission vector of device i with $u_{i,j,t} = 1$ implying that device i will exchange its local model with device j at iteration t, and $u_{i,j,t} = 0$, otherwise. $||\boldsymbol{u}_{i,t}|| = \sum_{j\in\mathcal{M}} u_{i,j,t}$ is the number of devices that will transmit the updated parameters to device i. The subset of devices selected by device i with which it shares its updated parameter $\mathcal{M}_{i,t} = \{j|u_{i,j,t} = 1\}$.

C. DFL Model Transmission Process

We adopt an orthogonal frequency division multiple access (OFDMA) transmission scheme [16]. Let W be the bandwidth that the device can use for model parameter transmission and $p_{i,j,t}$ be the transmit power. The data transmission delay of device i transmitting updated parameters to device j is

$$l_{i,j,t}\left(Q_{i,t}, \boldsymbol{u}_{i,t}, \boldsymbol{\varepsilon}_t, p_{i,j,t}\right) = \frac{Q_{i,t}}{\frac{W}{||\boldsymbol{u}_{i,t}||}\log\left(1+\frac{p_{i,j,t}h_{i,j,t}(\boldsymbol{\varepsilon}_t)}{\sigma_N^2}\right)}, \tag{10}$$

where $h_{i,j,t} = \rho_{i,j,t}d_{i,j,t}^{-2}$ is the channel gain between device i and j with $\rho_{i,j,t}$ being the Rayleigh fading parameters, and $d_{i,j,t}$ is the distance between device i and j. The location of each device i at time t is captured by a vector $\boldsymbol{\varepsilon}_{i,t} = [\varepsilon_{i,t,1}, \varepsilon_{i,t,2}]$, with $\boldsymbol{\varepsilon}_t = [\boldsymbol{\varepsilon}_{1,t}, ..., \boldsymbol{\varepsilon}_{M,t}]$ being the location matrix across users at iteration t. σ_N^2 represents the variance of additive white Gaussian noise.

D. Problem Formulation

Our goal is to minimize the DFL training loss across various downstream tasks while accounting for transmission delay and computational resource constraints. The optimization problem is formulated as

$$\min_{\boldsymbol{U},\boldsymbol{P},\boldsymbol{M}^A_{i,t},\boldsymbol{M}^B_{i,t}} \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{M} F\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \mathcal{D}_j\right), \tag{11}$$

$$\text{s.t. } l_{i,j,t}\left(Q_{i,t}, \boldsymbol{u}_{i,t}, \boldsymbol{\varepsilon}_t, p_{i,j,t}\right) \leqslant \Gamma, \forall i, j \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{11a}$$

$$\sum_{j\in\mathcal{M}_{i,t}} p_{i,j,t} \leqslant p_{\max}, \forall i \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{11b}$$

$$c_{i,t} \leqslant C_{max}, \forall i \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{11c}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_T]^\top$ is the FL model transmission matrix, and $\boldsymbol{P} = [\boldsymbol{p}_1, \cdots, \boldsymbol{p}_T]^\top$ is the transmit power matrix. $p_{\max}$ is the transmit power constraint. Γ is the maximum model transmission delay. (11a) is a constraint on the model transmission delay per iteration, (11b) is the transmit power constraint, and (11c) is the computing overhead constraint.

The problem in (11) is challenging to solve by traditional DFL algorithms due to the following reasons. First, the direct aggregation of the fine-tuned matrices $\boldsymbol{A}_{i,t}$ and $\boldsymbol{B}_{i,t}$ that come from devices with various downstream tasks will cause knowledge conflicts and prevent multi-task LLMs from converging, which calls for an efficient knowledge interference-free method for DFL. Second, the design of $\boldsymbol{U}$ and sparsity masks is challenging, and their interplay with DFL performance needs to be investigated.

III. Orthogonal and Sparse Activations based Multi-tasks Fine-tuning in DFL

To solve problem (11), we propose a fully distributed DFL fine-tuning algorithm that includes an orthogonal-LoRA that does not rely on other devices' parameters, as well as a sparse-activation based light-weight method considering parameter's importance. Then, we discuss the impact of the device connection topology $\boldsymbol{U}$ on the DFL performance under the designed sparse-and-orthogonal LoRA. Finally, we propose a cluster-based device topology design approach to accelerate the DFL performance under resource constraint.

A. Orthogonal-based Interference-free Multi-tasks Fine-tuning Method

According to [12], keeping the projection matrix $\boldsymbol{A}_{i,t}$ orthogonality between devices (i.e., $\sum ||\boldsymbol{A}_{i,t}\boldsymbol{A}_{i,t}^\top||^2 \approx 0, \forall i, j \in \mathcal{M}$) can effectively reduce knowledge interference between sub-tasks' updates. However, the orthogonality between $\boldsymbol{A}_{i,t}, \forall i \in \mathcal{M}$ will be compromised during each local fine-tuning process, as shown in (3). Hence, traditional orthogonality methods need to collect updated parameters from all other devices to keep this property at each iteration. In this scenario, to satisfy the orthogonality between $\boldsymbol{A}_{i,t}, \forall i \in \mathcal{M}$ in DFL without collecting parameters from other devices, we regard $\boldsymbol{A}_{i,t}$ as a static matrix with entries drawn from a zero-mean, unit-variance Gaussian distribution independently, inspired by LoRI [17]. In this way, $\boldsymbol{A}_{i,t}, \forall i \in \mathcal{M}$ are mutually orthogonal during the fine-tuning process, which is given by

$$\sum ||\boldsymbol{A}_{i,t}^\top \boldsymbol{A}_{j,t}||^2 \approx \boldsymbol{0}_{r\times r}, [\boldsymbol{A}_i]_{x,y} \sim \mathcal{N}(0,1), \forall i, j \in \mathcal{M}. \tag{12}$$

To maintain $\boldsymbol{A}_{i,t}$ as a static matrix, we also exclude it from the aggregation process outlined in Eq. (8), which is a standard step in traditional DFL. Hence, $\boldsymbol{A}_{i,t}$ is time-invariant and can be abbreviated as $\boldsymbol{A}_i$. Keeping $\boldsymbol{A}_i$ static does not compromise

the fine-tuning accuracy compared to traditional approaches where $\boldsymbol{A}_i$ is continuously updated, as it preserves the same subspace dimensionality. Furthermore, as $\boldsymbol{A}_i$ is excluded from the aggregation process, it only needs to be transmitted once across different devices, which can reduce the transmission overhead.

B. Sparse-activation based Light-weight method

To maximize the fine-tuning performance with sparse activated $\boldsymbol{B}_{i,t}$ with mask $\boldsymbol{M}_{i,t}^B$ generation method, where the parameters with highest $s_{\boldsymbol{M}_{i,t}^B,t}$ (abbreviated as $s_{i,t}$ in the following section) value is activated during fine-tuning, which can be given by:

$$\left[\boldsymbol{M}_{i,t}^B\right]_{x,y} = \mathbb{1}\left(\left[\boldsymbol{B}_{i,t}\right]_{x,y} \geq \tau_{s_{i,t}}\right), \tag{13}$$

where $\mathbb{1}_{\{x\}} = x$ if $x > 0$, $\mathbb{1}_{\{x\}} = 0$, otherwise. $\tau_{s_{i,t}}$ is the threshold value of the largest $s_{i,t}\%$ value. Thus, the updated adapter of device i can be given by

$$\boldsymbol{w}_{i,t} = \boldsymbol{A}_i\left(\boldsymbol{B}_{i,t} \odot \boldsymbol{M}_{i,t}^B\right). \tag{14}$$

We assume a subset $\mathcal{M}_{i,t} \subseteq \mathcal{M}$ of devices aggregate their $\boldsymbol{B}_{i,t}$ based on (9) and exchanged $\boldsymbol{A}_i$, the overall adapter can be given by

$$\boldsymbol{w}_{i,t} = \sum_{j\in\mathcal{M}_{i,t}} \alpha_j \boldsymbol{A}_j \left(\frac{1}{|\mathcal{M}_{i,t}|}\sum_{k\in\mathcal{M}_{i,t}} \boldsymbol{B}_{k,t} \odot \boldsymbol{M}_{k,t}^B\right), \tag{15}$$

where α_j represents the adapter's importance in dealing with specific verification subtasks.

C. Device Connection and DFL Convergence Analysis

To simplify the device connection $\boldsymbol{U}$ optimization in (11), we must analyze the impact of $\boldsymbol{U}$ on the DFL model convergence under orthogonal $\boldsymbol{A}_i$ and sparse activation $\boldsymbol{B}_{i,t}$ design. Since the each projection matrix $\boldsymbol{A}_i$ is independently sampled by Gaussian distribution, and each expansion matrix $\boldsymbol{B}_{i,t}$ is updated based on independent dataset and sparse mask, each adapter $\boldsymbol{w}_{i,t}$ and the masks $M_{i,t}$ are independent with each other, we can first make the following assumptions [18]:

- *Assumption* 1*:* For device $i, j \in \mathcal{M}$, we assume $E||\boldsymbol{A}_i\nabla_{\boldsymbol{B}_{i,t}}F\left(\boldsymbol{W}_0, \boldsymbol{w}_{i,t}, \mathcal{D}_i\right)|| \leq G$.
- *Assumption* 2*:* For each device $i \in \mathcal{M}$, we assume $E||\frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i \sum_{j\in\mathcal{M}_{i,t}} \boldsymbol{B}_{j,t}|| \leq P$.

These assumptions are natural, where Assumption 1 stems from the fact that the upper bound of the local gradient exists, and Assumption 2 assumes an upper bound of the product between Gauss-sampled $\boldsymbol{A}_i$ and independent updated $\boldsymbol{B}_{j,t}$ exists. We also define the parameter collision rate $S_{\mathcal{M}_{i,t}} \in [0,1]$ caused by aggregating $\boldsymbol{B}_{i,t}\odot\boldsymbol{M}_{i,t}^B$ from devices in $\mathcal{M}_{i,t}$ based on the independence sparsity rate $s_{i,t}$, which can be given by

$$S_{\mathcal{M}_{i,t}} = 1 - \prod_{j\in\mathcal{M}_{i,t}}(1-s_{j,t}) - \sum_{j\in\mathcal{M}_{i,t}} s_{j,t}\prod_{z\in\mathcal{M}_{i,t},z\neq j}(1-s_{z,t}). \tag{16}$$

The parameter collision rate $S_{\mathcal{M}_{i,t}}$ represents the parameters' overlap probability between devices' independence masks, which depends on the device connection topology $\boldsymbol{U}$. Based on the definitions, we analyze the impact of $\boldsymbol{U}$ on the upper bound of the gap between the local fine-tuned adapter and the aggregated global adapter, which can be given by

Theorem 1. Given the model transmission matrix $\boldsymbol{U}$, an upper bound of the gap between each local model $\boldsymbol{A}_i\left(\boldsymbol{B}_{i,t}\odot\boldsymbol{M}_{i,t}\right)$ of device i and the average model $\frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i\sum_{j\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\odot\boldsymbol{M}_{j,t}$ of all devices in $\mathcal{M}_{i,t}$ can be given by

$$\sum_{i\in\mathcal{M}_{i,t}} E||\boldsymbol{A}_i\left(\boldsymbol{B}_{i,t}\odot\boldsymbol{M}_{i,t}\right) - \frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i\sum_{j\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\odot\boldsymbol{M}_{j,t}||^2 \leq 2rk\sum_{i=1}^{M}\sum_{j=1}^{M} S_{\mathcal{M}_{i,t}}(G+P). \tag{17}$$

Proof. See Appendix A. □

From Theorem 1, we can see that the gap between each local adapter of device i and the average adapters of all devices in $\mathcal{M}_{i,t}$ is affected by $S_{\mathcal{M}_{i,t}}$, which depends on the device connection topology $\boldsymbol{U}$. From Theorem 1, we can also observe that this gap decreases as the parameter collision rate $S_{\mathcal{M}_{i,t}}$ decreases, which implies that the device connection topology design that can satisfy the parameter collision-free condition between $\boldsymbol{B}_{i,t}$ can effectively enhance the DFL performance. This is due to the fact that each device updates $\boldsymbol{B}_{i,t}\odot\boldsymbol{M}_{i,t}$ with activation overlap simultaneously will cause a multi-party tug-of-war, leading to inconsistencies in the optimization directions, thus reducing the DFL performance. Furthermore, since the parameter collision rate $S_{\mathcal{M}_{i,t}}$ grows with the number of devices in $\mathcal{M}_{i,t}$, it is essential to perform aggregation separately.

D. Cluster-based Device Connection Topology Design

To enhance DFL performance, we propose a cluster-based method where devices are grouped into clusters, and the aggregation of $\boldsymbol{B}_{i,t}$ is performed separately within each cluster (named inner-cluster aggregation), thereby guaranteeing $S_{\mathcal{M}_{i,t}}$ does not exceed the threshold $S_{\max}$. After iterations of inner-cluster aggregation, inter-cluster parameter exchange is performed, which only exchanges parameters without further aggregation, and hence, can expand the scope of knowledge and avoid parameter collision between clusters fundamentally. In particular, we employ an AGNES-based cluster algorithm [19] considering devices' computing, communication resources, and parameter collision rate constraints, where each device is initially considered as a single-element cluster (leaf). At each step of the algorithm, neighbor clusters exchange communication and sparsity rate with each other and find those clusters with the lowest parameter collision rate, as well as satisfy the transmission power constraint in (11b), and combine into a new, bigger cluster. This procedure is iterated until all devices are members of just one single big cluster or until the pre-defined parameter collision rate $S_{\max}$ is reached.

The parameter aggregation and exchange in the proposed cluster-based algorithm can be summarized as follows:

1) Devices aggregate into different clusters $\mathcal{M}_{i,t} \subseteq \mathcal{C}$ based on cluster algroithm.
2) Each device i perform local fine-tuning and exchange parameter $\boldsymbol{A}_i$ and $\boldsymbol{B}_{i,t}$ with other devices available in cluster for aggregation.
3) Each device in cluster $\mathcal{M}_{i,t}$ store the received $\boldsymbol{A}_i$ and aggregate the received $\boldsymbol{B}_{i,t}$ based on (9).
4) After iterations of updating, clusters exchange the stored $\boldsymbol{A}_i$ and $\boldsymbol{B}_{i,t}$ with each other for knowledge sharing.

Steps 2)-4) are performed until the DFL model convergence.

Hence, each device keeps a group of adapters generated by different clusters $\mathcal{C} = \{\mathcal{M}_{i,t}\}$, the adapter used for the sub-task can be given by

$$\boldsymbol{w}_t = \sum_{\mathcal{M}_{i,t} \subseteq \mathcal{C}} \sum_{j \in \mathcal{M}_{i,t}} \alpha_j \boldsymbol{A}_j \left(\frac{1}{|\mathcal{M}_{i,t}|} \sum_{j \in \mathcal{M}_{i,t}} \boldsymbol{B}_{j,t} \odot \boldsymbol{M}_{j,t} \right), \tag{18}$$

where α_j is the affinity score of the adapter $\boldsymbol{w}_{j,t}$.

IV. Simulation Results

For our simulations, we consider a DFL framework with a circular area and uniformly distributed $M \leq 10$ devices across the region, each possessing heterogeneous computing and communication capabilities. Each device is equipped with the same pre-trained Qwen 2.5-1.5B-Instruct LLM and a unique fine-tuning dataset that focuses on four capabilities: (i) natural language understanding, which includes BoolQ, Piqa, and SocialIqa, (ii) mathematical reasoning, which includes GSM8K, Arc-easy, Arc-challenge, (iii) code generation, which includes CodeAlpaca, HumanEval, and (iv) comprehensive ability, which includes DollyTails, Hellaswag, and OpenBookQA. For comparison, we utilize two baselines :

- A decentralized fine-tuning method based on parameter orthogonalization and a sparsification method in LoRI [17] where devices generate a random device connection under communication and computing resources constraint (labeled "LoRI" in plots).
- A decentralized fine-tuning based on traditional LoRA [9] (without parameter orthogonalization and sparsification) and a random device connection under communication resources constraint (labeled "LoRA" in plots).

In Fig. 2, we show how the average parameter collision rate changes as the number of devices varies. From Fig. 2, we can observe that the proposed method achieves the lowest average parameter collision rate, which is strictly constrained by the predefined threshold $S_{\max}$, whereas the other baselines exhibit higher parameter collision rates. This is due to the fact that the proposed algorithm reduces the parameter collision rate through cluster-based DFL device connection topology design, which considers each device's sparsity rate and the number of devices in the cluster. In contrast, the average parameter collision rate of LoRI increases as the number of devices

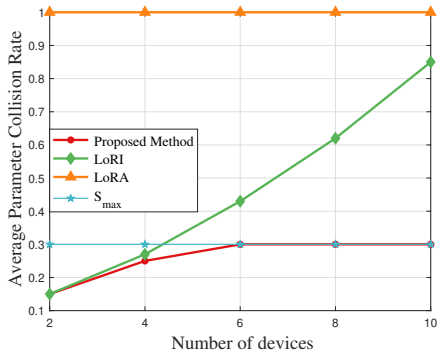

Fig. 2. Average Parameter Collision Rate vs. the Number of Devices.

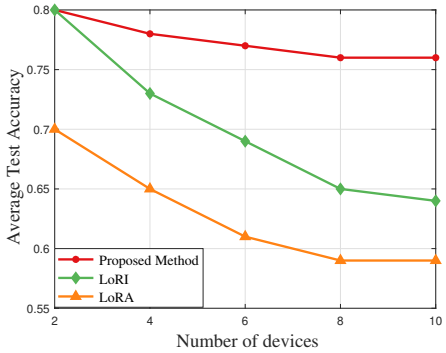

Fig. 3. Test accuracy vs. the Number of Devices.

participating in the DFL increases, while LoRA, lacking parameter sparsity, consistently experiences a 100% parameter collision rate. In Fig. 3, we illustrate how the average test accuracy of tasks changes as the number of devices changes. From Fig. 3, we can see that the average test accuracy of the proposed method decreases slightly and plateaus as the number of users increases. In contrast, the average test accuracy of LoRI and LoRA decreases rapidly as the number of devices increases. In particular, compared to LoRI and LoRA based method, the proposed method can improve the average test accuracy by up to 12% and 22%, respectively. This is due to the fact that as the number of devices increases, the proposed method can maintain an acceptable parameter collision rate through cluster-based device connection topology design while other baselines exhibit an increase in parameter collision rate with more devices, as shown in Fig. 2.

In Table I, we compare the proposed method against baselines in terms of the number of trainable parameters and performance across 8 tasks (only 3 of them are shown in the table). From table I, we can observe that the proposed method can reduce the trainable parameter that needs to be updated and transmitted compared to LoRA by up to 55%. This is due to the fact that in the proposed method, $\boldsymbol{A}_i$ is kept static during fine-tuning, which does not need to be updated and transmitted, which can reduce the computing and communication overhead.

TABLE I
PERFORMANCE AND TRANSMISSION OVERHEAD COMPARISON.

Method	# Params (%)	GSM8K	BoolQ	Hellaswag
Proposed Method	15M (0.48%)	77%	65%	79%
LoRI	16.5M (0.54%)	73%	59%	70%
LoRA	33.8M (1.1%)	70%	51%	65%

V. CONCLUSION

In this paper, we introduced a novel sparse-and-orthogonal LoRA method to reduce the computational and communication resources consumption, as well as reducing knowledge interference between heterogeneous datasets. Based on the proposed methods, we further analyzed and concluded that the performance of DFL is directly related to the parameter collision rate, which is affected by the device connection topology. Thus, we designed a cluster-based device connection topology design method to reduce the parameter collision rate by jointly considering parameter orthogonal and sparse activations. Simulation results demonstrated that the proposed algorithm can achieve robust DFL compared to traditional LoRA and LoRI methods, while ensuring lower computational and communication resource consumption.

VI. APPENDIX

A. Proof of Theorem 1

Under assumptions, we have

$$\begin{aligned}
&\sum_{i\in\mathcal{M}_{i,t}} E||\boldsymbol{A}_i\boldsymbol{B}_{i,t}\odot\boldsymbol{M}_{i,t}-\frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i\sum_{j\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\odot\boldsymbol{M}_{j,t}||^2\\
\leq&\sum_{i\in\mathcal{M}_{i,t}} 2E||\boldsymbol{A}_i\boldsymbol{B}_{i,t}\odot\boldsymbol{M}_{i,t}-\frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i\sum_{j\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\odot\boldsymbol{M}_{i,t}||\\
&+\sum_{i\in\mathcal{M}_{i,t}} 2E||\frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i\sum_{j\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\odot\boldsymbol{M}_{i,t}\\
&-\frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i\sum_{j\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\odot\boldsymbol{M}_{j,t}||.
\end{aligned}\tag{19}$$

For the first term in (19), we have

$$\begin{aligned}
&\sum_{i\in\mathcal{M}_{i,t}} E||\boldsymbol{A}_i\boldsymbol{B}_{i,t}\odot\boldsymbol{M}_{i,t}-\frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i\sum_{j\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\odot\boldsymbol{M}_{i,t}||\\
=&\sum_{i\in\mathcal{M}_{i,t}} E||\boldsymbol{A}_i\frac{1}{|\mathcal{M}_{i,t}|}\sum_{j\in\mathcal{M}_{i,t}}(\boldsymbol{B}_{i,t}-\boldsymbol{B}_{j,t})\odot\boldsymbol{M}^c_{i,j,t}||.
\end{aligned}\tag{20}$$

Due to the definition of parameter collision rate, (20) can be further given by

$$\begin{aligned}
&rk\sum_{i\in\mathcal{M}_{i,t}}\sum_{j\in\mathcal{M}_{i,t}} S_{\mathcal{M}_{i,t}}E||\boldsymbol{A}_i\left(\boldsymbol{B}_{i,t}-\frac{1}{|\mathcal{M}_{i,t}|}\sum_{j\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\right)||\\
=&rk\sum_{i\in\mathcal{M}_{i,t}}\sum_{j\in\mathcal{M}_{i,t}} S_{\mathcal{M}_{i,t}}E||-\eta\boldsymbol{A}_i\nabla_{\boldsymbol{B}_{i,t}}F_i||\\
\leq&rk\sum_{i\in\mathcal{M}_{i,t}}\sum_{j\in\mathcal{M}_{i,t}} S_{\mathcal{M}_{i,t}}G
\end{aligned}\tag{21}$$

The second equation is due to the fact that $\boldsymbol{A}_i$ is independent to $\frac{1}{|\mathcal{M}_{i,t}|}\sum_{j\in\mathcal{M}_{i,t}}(\boldsymbol{B}_{i,t}-\boldsymbol{B}_{j,t})\,|\boldsymbol{M}_{j,t}-\boldsymbol{M}_{i,t}|\odot\boldsymbol{M}_{i,t}$. Similarly, we can rewrite the second term in (19) as

$$\begin{aligned}
&\sum_{i\in\mathcal{M}_{i,t}} E||\frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i\sum_{j\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\odot\boldsymbol{M}_{j,t}\\
&-\frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i\sum_{j\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\odot\boldsymbol{M}_{i,t}||\leq rk\sum_{i\in\mathcal{M}_{i,t}} S_{\mathcal{M}_{i,t}}P
\end{aligned}\tag{22}$$

Thus, we have

$$\begin{aligned}
&\sum_{i\in\mathcal{M}_{i,t}} E||\boldsymbol{A}_i\boldsymbol{B}_{i,t}\odot\boldsymbol{M}_{i,t}-\frac{1}{|\mathcal{M}_{i,t}|}\boldsymbol{A}_i\sum_{i\in\mathcal{M}_{i,t}}\boldsymbol{B}_{j,t}\odot\boldsymbol{M}_{j,t}||^2\\
&\leq 2rk\sum_{i\in\mathcal{M}_{i,t}}\sum_{j\in\mathcal{M}_{i,t}} S_{\mathcal{M}_{i,t}}(G+P)
\end{aligned}\tag{23}$$

This ends the proof.

REFERENCES

[1] K. B. Kan, H. Mun, G. Cao, and Y. Lee, "Mobile-LLaMA: Instruction Fine-Tuning Open-Source LLM for Network Analysis in 5G Networks," *IEEE Network*, vol. 38, no. 5, pp. 76–83, July 2024.

[2] Y. Su, Y. Du, Y. Deng, and M. Dohler, "Towards Communication Efficient Multi-Agent Cooperations: Reinforcement Learning and LLM," *IEEE Transactions on Vehicular Technology*, pp. 1–15, Oct. 2025.

[3] Y. Huang, H. Du, X. Zhang, D. Niyato, J. Kang, Z. Xiong, S. Wang, and T. Huang, "Large Language Models for Networking: Applications, Enabling Techniques, and Challenges," *IEEE Network*, vol. 39, no. 1, pp. 235–242, July 2024.

[4] D. Wei, X. Xu, Y. Liu, H. V. Poor, and M. Chen, "Optimizing model splitting and device task assignment for deceptive signal assisted private multi-hop split learning," *IEEE Journal on Selected Areas in Communications*, vol. Early Access, Nov. 2025.

[5] Y. Yang, Y. He, B. Zhao, C. Wu, Z. Gao, and L. Rui, "Multirepresentation spatial-temporal graph convolutional networks for network traffic prediction," *IEEE Internet of Things Journal*, vol. 12, no. 13, pp. 23085–23099, Mar. 2025.

[6] Y. He, Y. Yang, B. Zhao, Zhipeng Gao, and Lanlan Rui, "Network traffic prediction method based on multi-channel spatial-temporal graph convolutional networks," in *IEEE International Conference on Advanced Infocomm Technology (ICAIT)*, Chongqing, China, Aug. 2022.

[7] X. Yi, C. Hu, B. Cai, H. Huang, Y. Chen, and K. Wang, "FedALoRA: Adaptive Local LoRA Aggregation for Personalized Federated Learning in LLM," *IEEE Internet of Things Journal*, June 2025.

[8] N. Yan, Y. Su, Y. Deng, and R. Schober, "Federated Fine-Tuning of LLMs: Framework Comparison and Research Directions," *IEEE Communications Magazine*, vol. 63, no. 10, pp. 52–58, Sep. 2025.

[9] P. Wallis Z. Allen-Zhu Y. Li S. Wang L. Wang W. Chen E. J. Hu, Y. Shen, "LoRA: Low-Rank Adaptation of Large Language Models," vol. abs/2106.09685, 2021.

[10] B. Kim and W. Choi, "Communication-Efficient Wireless Federated Fine-Tuning for Large-Scale AI Models," *ArXiv*, vol. abs/2505.00333, 2017.

[11] S. Yang, K. Ning, Y. Liu, J. Yao, Y. Tian, Y. Song, and L. Yuan, "Is Parameter Collision Hindering Continual Learning in LLMs?," vol. arXiv:2410.10179, Dec. 2024.

[12] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal Gradient Descent for Continual Learning," *ArXiv*, vol. abs/1910.07104, Oct. 2019.

[13] Y.-S. Liang and W.-J. Li, "Adaptive Plasticity Improvement for Continual Learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, Aug. 2023.

[14] S. Ghiasvand, M. Alizadeh, and R. Pedarsani, "Decentralized Low-Rank Fine-Tuning of Large Language Models," *ArXiv*, vol. arXiv:2501.15361, Aug. 2025.

[15] N. Yang, S. Wang, Y. Liu, C. G. Brinton, C. Yin, and M. Chen, "Graph neural networks for the optimization of collaborative federated learning energy efficiency," *IEEE Transactions on Mobile Computing*, vol. 24, no. 11, pp. 12049–12060, June 2025.

[16] X. Xu, H. Xu, D. Wei, W. Saad, M. Bennis, and M. Chen, "Transformer based Collaborative Reinforcement Learning for Fluid Antenna System (FAS)-enabled 3D UAV Positioning," *IEEE Journal on Selected Areas in Communications*, vol. Early Access, Oct. 2025.

[17] A. Panda T. Goldstein J. Zhang, J. You, "LoRI: Reducing Cross-Task Interference in Multi-Task Low-Rank Adaptation," *Conference on Language Modeling (COLM)*, 2025.

[18] S. Wang, H. Guo, X. Zhu, C. Yin, and V. K. N. Lau, "Communication-Efficient Distributed Bayesian Federated Learning Over Arbitrary Graphs," *IEEE Transactions on Signal Processing*, vol. 73, pp. 1351–1366, Feb. 2025.

[19] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, New York, USA, 1990.