EXPLORING SELECTIVE LAYER FINE-TUNING IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) has emerged as a promising paradigm for fine-tuning foundation models using distributed data in a privacy-preserving manner. Under limited computational resources, clients often find it more practical to fine-tune a selected subset of layers, rather than the entire model, based on their task-specific data. In this study, we provide a thorough theoretical exploration of selective layer fine-tuning in FL, emphasizing a flexible approach that allows the clients to adjust their selected layers according to their local data and resources. We theoretically demonstrate that the layer selection strategy has a significant impact on model convergence in two critical aspects: the importance of selected layers and the heterogeneous choices across clients. Drawing from these insights, we further propose a strategic layer selection method that utilizes local gradients and regulates layer selections across clients. Extensive experiments on both image and text datasets demonstrate the effectiveness of the proposed strategy compared with several baselines, highlighting its advances in identifying critical layers that adapt to the client heterogeneity and training dynamics in FL.

1 INTRODUCTION

Foundation models (Bommasani et al., 2021), including BERT (Devlin et al., 2019), GPT (Radford et al., 2019; Brown et al., 2020), CLIP (Radford et al., 2021; Dosovitskiy et al., 2021), LLaMA (Touvron et al., 2023), and so on (Ramesh et al., 2021; Chowdhery et al., 2023), have attracted considerable attention due to their exceptional ability in handling complex tasks (Eloundou et al., 2023). When it comes to practical deployments of these models in specialized fields, fine-tuning with domain-specific data becomes critical. Nevertheless, the distributed nature of data across various users and organizations presents a challenge for centralized storage and training, as it may lead to severe privacy concerns and incur additional transmission costs. Such issues have positioned federated learning (FL) (McMahan et al., 2017) as a promising paradigm to fine-tune foundation models, aligning model enhancement with privacy preservation (Chen et al., 2023a; Kuang et al., 2023).

Generally, FL aims to learn a global model through a collaborative process where clients perform local training and upload the parameter updates to a central server for aggregation. Given that clients have limited resources (Bonawitz et al., 2019; Imteaj et al., 2022), such as computational power, communication bandwidth, and available memory, it becomes impractical for them to fine-tune the entire foundation model. Two kinds of solutions have recently emerged to tackle this challenge. The first solution employs *parameter-efficient fine-tuning* techniques (Houlsby et al., 2019; Gao et al., 2021; Hu et al., 2022; Li & Liang, 2021), which introduces additional modules integrated into foundation models and updates these modules with domain-specific data while keeping the parameters of the foundation model frozen. The second one is *selective model fine-tuning* (Lee et al., 2019a; Xu et al., 2021; Zhang et al., 2022a; Shen et al., 2021), which only selects an impactful subset of parameters for optimization to streamline the fine-tuning process under resource constraints.

This study focuses on selective model fine-tuning as it is particularly well-suited to address the inherent heterogeneity in FL, i.e., the data heterogeneity and device heterogeneity (Yang et al., 2021; Chai et al., 2019; Li et al., 2022). Specifically, clients involved in FL have non-independent and identically distributed (non-IID) data and different system resources, leading to the need to customize fine-tuning strategies to such discrepancies. For example, clients with limited computation resources may opt to update only a fraction of the model, while those with sufficient resources and

high-quality data prefer fine-tuning a large portion of the model to enhance performance. Selective model fine-tuning enables clients to adjust the chosen part of the model to be updated based on their own capabilities, providing a flexible and advanced solution to mitigating sub-optimal issues induced by the heterogeneity in FL.

The exploration of selective model fine-tuning within the context of FL, is still in its early stages. Previous studies (Shen et al., 2021; Xu et al., 2021; Lee et al., 2022; Dun et al., 2022) have concentrated on designing static strategies for subnetwork selection to improve model fine-tuning performance, without adequately considering heterogeneity among clients. To fulfill this gap, in this paper, we provide a comprehensive theoretical analysis on selective model fine-tuning in FL, focusing on a general scenario where clients are allowed to choose different layers for local training and vary their choices across different training epochs, called *selective layer fine-tuning*. Specifically, we formulate the optimization objective of selective layer fine-tuning in FL, and provide insights on effectively determining critical layers to achieve model convergence. Building on these insights, we further propose a novel layer selection strategy that leverages local gradients and the regulation of unified selections.

Our main contributions are summarized as follows:

- We study a practical FL setup where clients choose to fine-tune some layers of the model, with arbitrary layer selection that may vary among clients and across different training epochs. We theoretically analyze such a training scheme and investigate the impact of layer selection. The analytical results show that the selected layers affect the convergence performance with two critical aspects, namely the importance of layers and heterogeneous choices across clients.
- Building on the theoretical analysis, we formulate the optimization problem of selective layer fine-tuning considering the limited and diverse resource budgets of clients. Inspired by the solution to this optimization problem, we propose an effective strategy for selecting layers for fine-tuning that are well-suited for the local data and available resources at clients.
- We conduct experiments to compare the proposed layer selection strategy with baseline methods on both image and text datasets. Experimental results demonstrate the superior performance of the proposed strategy in achieving better model performance, highlighting that the proposed strategy can find critical layers for fine-tuning while considering the client heterogeneity and training dynamics in FL[1].

2 Related Works

Various approaches have been proposed to properly select a subset of parameters for fine-tuning foundation models within centralized training, including optimizing a non-structured mask matrix (Lee et al., 2019a; Xu et al., 2021; Zhang et al., 2022a; Shen et al., 2021; Zaken et al., 2022; Zhang et al., 2023; Kovaleva et al., 2019; Lee et al., 2019b) and adopting layer-wise selection strategies (Kovaleva et al., 2019; Lee et al., 2019b; 2022; Kaplun et al., 2023). For example, Lee et al. (2019a) suggest updating the parameters in a stochastic manner based on the Bernoulli distribution, while Kovaleva et al. (2019); Lee et al. (2019b) showcase that fine-tuning the top few layers achieves competitive model performance in downstream tasks. Moreover, Lee et al. (2022) propose to select layers according to their gradient statistics.

Recent studies have extended the selective fine-tuning techniques to FL scenarios (Nguyen et al., 2022a; Chen et al., 2022; Hilmkil et al., 2021; Zhang et al., 2022b). Specifically, researchers (Lee et al., 2023; Dun et al., 2022) investigate layer-wise network decomposition to achieve selective model fine-tuning on clients. However, these works fail to offer methodologies for adaptive and dynamic layer selection that takes into account the heterogeneous characteristics of clients. In addition, personalized FL algorithms (Pillutla et al., 2022; Chen et al., 2023b) propose to train different subnetworks on clients towards better local models. Different from previous studies, we focus on providing an in-depth understanding of selective layer fine-tuning in FL, considering the heterogeneity from the perspective of client resources and local data distributions.

[1]The source codes are available at `https://anonymous.4open.science/r/fed_selected_tune/`.

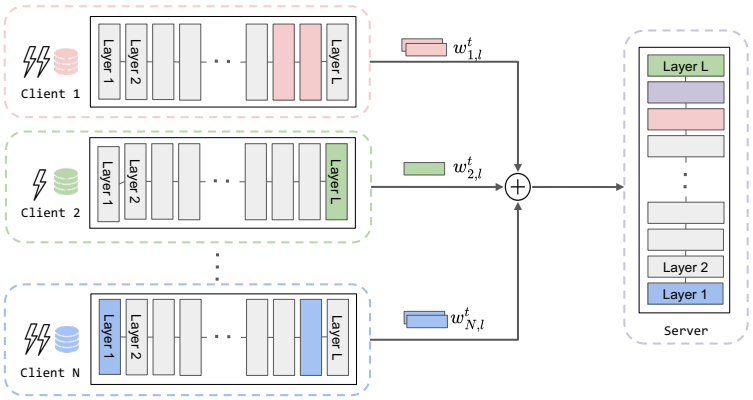

Figure 1: An overview of selective layer fine-tuning in FL. The colored layers are selected for fine-tuning.

3 Problem Formulation

Federated learning We consider an FL system with a central server and N clients (denoted by the set $\mathcal{N} = \{1, \ldots, N\}$), where each client has a private dataset $\mathcal{D}_i$ consisting of $d_i = |\mathcal{D}_i|$ data instances. The server owns a pretrained foundation model $\theta \in \mathbb{R}^P$, containing P trainable parameters and L layers with the index set $\mathcal{L} = \{1, 2, \ldots, L\}$. The server aims to fine-tune this foundation model based on the clients' datasets $\mathcal{D} = \{\mathcal{D}_i\}_{i \in \mathcal{N}}$ but does not directly access these datasets. The learning goal is formally given as:

$$\min_{\theta \in \mathbb{R}^P} f(\theta) = \sum_{i=1}^{N} \alpha_i f_i(\theta), \tag{1}$$

where $\alpha_i = \frac{d_i}{\sum_{j=1}^{N} d_j}$ denotes the relative sample size, and $f_i(\theta) = \frac{1}{d_i} \sum_{\xi \in \mathcal{D}_i} F_i(\theta; \xi)$ denotes the local training objective of client i. Here we use $F_i(\theta; \xi)$ to define the (possibly non-convex) loss function computed by the model θ on data sample ξ. The training process of FL is divided into T training epochs. In each epoch $t \in [T]$, the server chooses a subset of clients $\mathcal{S}^t$, and sends the up-to-date global model θ^t to these clients for local training.

Selective layer fine-tuning in FL An overview of selective layer fine-tuning in FL is illustrated in Figure 1. Due to resource limitations, clients tend to update some of the layers in the local training process rather than the entire global model. Formally, we define a masking vector $\mathbf{m}_i^t \in \{0, 1\}^L$ for each client i. The l-th element $\mathbf{m}_i^t(l)$ equals 1 if the l-th layer is selected to be updated in the t-th training epoch, and $\mathbf{m}_i^t(l) = 0$ otherwise. Accordingly, the selected layer set of client i is denoted by $\mathcal{L}_i^t \triangleq \{l \in \mathcal{L} | \mathbf{m}_i^t(l) = 1\}$, and the set for all selected layers in the t-th training epoch is denoted by $\mathcal{L}_t = \bigcup_{i \in \mathcal{S}^t} \mathcal{L}_i^t$. The choice of selected layer sets has a substantial effect on training performance, which will be discussed in detail later.

After determining the selected layer set $\mathcal{L}_i^t$, clients initialize the local model according to the global model sent by the server, i.e., $\theta_i^{t,0} = \theta^t$, and train the local model for τ local steps using the mini-batch SGD algorithm (McMahan et al., 2017; Wang et al., 2020; Karimireddy et al., 2020). For local step $k \in [\tau]$, client i samples a batch of data instances $\xi_i^{t,k}$, and calculates the gradients for the selected layers, which is given as[2]:

$$\sum_{l \in \mathcal{L}_i^t} g_{i,l}(\theta_i^{t,k}; \xi_i^{t,k}) = \sum_{l \in \mathcal{L}_i^t} \nabla_l F_i(\theta_i^{t,k}; \xi_i^{t,k}). \tag{2}$$

Notably, the local gradient calculation pertains solely to the layers within the subset $\mathcal{L}_i^t$. Afterward, the local model is updated with the learning rate η:

$$\theta_i^{t,k} = \theta_i^{t,k-1} - \eta \sum_{l \in \mathcal{L}_i^t} g_{i,l}(\theta_i^{t,k-1}; \xi_i^{t,k-1}), \forall k \in \{1, 2, \ldots, \tau\}. \tag{3}$$

[2] $\nabla_l F(\theta)$ represents the gradient of a function $F(\theta)$ w.r.t. the parameters of the l-th layer in model θ.

Algorithm 1 Selective Layer Fine-tuning in FL

Input: The pre-trained global model θ^0
for $t = 0, 1, \ldots, T-1$ **do**
  Sample a set of clients $\mathcal{S}^t$;
  Broadcast the up-to-date global model θ^t and selected layer set $\mathcal{L}_i^t$ to clients $\mathcal{S}^t$;
  for each client i in $\mathcal{S}^t$ **do**
    Compute the gradients w.r.t. layers $\mathcal{L}_i^t$ and update the model for τ steps; {▷ Equation (3)}
    Upload the accumulated updates Δ_i^t to the server; {▷ Equation (4)}
  end for
  Compute the global update Δ^t; {▷ Equation (5)}
  Update the global model θ^t; {▷ Equation (6)}
end for
Return: The global model θ^T

The accumulated model update in local training is summarized as:

$$\Delta_i^t = \frac{1}{\eta}(\theta_i^{t,0} - \theta_i^{t,\tau}) = \sum_{k=0}^{\tau-1} \sum_{l \in \mathcal{L}_i^t} g_{i,l}(\theta_i^{t,k}; \xi_i^{t,k}). \tag{4}$$

After local training, clients upload their model updates $\Delta_i^t, i \in \mathcal{S}^t$ to the server. The server performs federated aggregation among these model updates and optimizes the global model accordingly:

$$\Delta^t = \sum_{l \in \mathcal{L}_t} \sum_{i \in \mathcal{S}^t} w_{i,l}^t \sum_{k=0}^{\tau-1} g_{i,l}(\theta_i^{t,k}; \xi_i^{t,k}), \tag{5}$$

and

$$\theta^{t+1} = \theta^t - \eta \Delta^t. \tag{6}$$

Inspired by previous studies (McMahan et al., 2017; Li et al., 2020), the aggregation weights in selective layer fine-tuning are defined based on the data ratio and the masking vectors, which are formally given as:

$$w_{i,l}^t = \begin{cases} \frac{d_i}{\sum_{\{j \in \mathcal{S}^t | \mathbf{m}_j^t(l)=1\}} d_j}, & \text{if } \mathbf{m}_i^t(l) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The details of the training process are summarized in Algorithm 1.

4 Which Layers Should Be Selected For Fine-Tuning?

The aforementioned training process provides substantial flexibility in selective layer fine-tuning, namely, clients are allowed to select different layers for local training and adjust their choices in different training epochs. Such flexibility enables clients to tailor their local training to their data and resources, providing feasible solutions for handling the heterogeneity in FL.

However, without a well-designed strategy for layer selection, the optimization of the global model in FL could be severely hindered, potentially leading to a suboptimal solution or even failure in convergence. As a result, researchers have proposed several useful strategies for layer selection in recent years, including:

- All clients select the same layer set for fine-tuning (Pillutla et al., 2022; Lee et al., 2019a; Zhang et al., 2022a; 2023; Lee et al., 2019b), i.e., $\mathcal{L}_i^t = \mathcal{L}_j^t, \forall i \neq j$;
- Clients fix their selections across different training epochs (Arivazhagan et al., 2019; Chen et al., 2023b), i.e., $\mathcal{L}_i^{t_1} = \mathcal{L}_i^{t_2}, \forall t_1, t_2 \in [T]$.

These strategies for layer selection are proposed based on the insights drawn from experts' experience, serving as special instantiations of the selected layer sets $\mathcal{L}_i^t$. It is worth noting that these experience-driven strategies might not consistently yield optimal results in various FL applications, particularly considering client heterogeneity. This leads to an essential question: *How to effectively determine the task-specified layer selection strategy among a large search space of possible options?* In the rest of this section, we provide a theoretical analysis to answer this question.

4.1 THEORETICAL ANALYSIS

Following previous theoretical analysis in FL (Wang et al., 2020; Karimireddy et al., 2020; Li et al., 2020), we begin with some necessary assumptions.

Assumption 4.1. *(γ-Smoothness) There exists a constant $\gamma > 0$ such that for any $\theta, \theta' \in \mathbb{R}^P$, $\|\nabla f_i(\theta) - \nabla f_i(\theta')\|_2 \leq \gamma \|\theta - \theta'\|_2, \forall i \in \mathcal{N}$.*

For analyzing the effect of each layer on the model convergence, we give several assumptions for the gradient with respect to each layer l.

Assumption 4.2. *(Unbiased and variance-bounded stochastic gradient) The stochastic gradient $g_{i,l}(\theta^t; \xi_i^t)$ on a randomly sampled batch of data ξ_i^t is an unbiased estimate of the full-batch gradient, i.e., $\mathbb{E}[g_{i,l}(\theta^t; \xi_i^t)] = \nabla_l f_i(\theta^t)$. Besides, there exist constants $\sigma_l > 0, \forall l \in \mathcal{L}$ such that $\|g_{i,l}(\theta^t; \xi_i^t) - \nabla_l f_i(\theta^t)\|^2 \leq \sigma_l^2, \forall i \in \mathcal{N}$ and $\sum_{l \in \mathcal{L}_t} \sigma_l^2 \leq \sigma^2$.*

The non-IID data owned by clients causes diverse gradients. In the following assumption, we state the diversity of each layer's gradient.

Assumption 4.3. *(Gradient diversity) There exist constants $\kappa_l > 0, \forall l \in \mathcal{L}$ such that $\|\nabla_l f(\theta^t) - \nabla_l f_i(\theta^t)\|^2 \leq \kappa_l^2, \forall i \in \mathcal{N}$.*

Here we first consider a case where $\tau = 1$ to simplify the analysis without affecting the insights on layer selection. The detailed analysis for the generalized case, i.e., $\tau > 1$, is provided in Appendix A.3.

Compared with the theoretical analysis for the standard FL settings (Wang et al., 2021; Li et al., 2020), there exist three additional challenges in selective layer fine-tuning. Firstly, since each client only updates some layers during the local training process, the aggregated gradient is no longer an unbiased estimate of the local gradient $\nabla f_i(\theta^t)$, i.e.,

$$\mathbb{E}[\Delta_i^t] = \sum_{l \in \mathcal{L}_i^t} \nabla_l f_i(\theta^t) \neq \nabla f_i(\theta^t), \tag{8}$$

where the inequality holds unless all layers are selected for fine-tuning, i.e., $\mathcal{L}_i^t = \mathcal{L}$. Secondly, since a certain layer may not be selected by all the clients, the aggregated gradient of this layer is not equivalent to the gradient computed based on the global loss function ($\sum_{l \in \mathcal{L}_t} \nabla_l f(\theta^t)$), which is given as:

$$\mathbb{E}[\Delta^t] = \sum_{l \in \mathcal{L}_t} \sum_{i \in \mathcal{S}^t} w_{i,l}^t \nabla_l f_i(\theta^t) \neq \sum_{l \in \mathcal{L}_t} \nabla_l f(\theta^t), \tag{9}$$

where the inequality holds unless all clients select the same subset of layers. Last but not least, the aforementioned gaps vary across different training epochs, making it rather complicated in the theoretical analysis.

To link the aggregated and desired gradients, we define a surrogate objective function representing the underlying loss function optimized by the clients, which is given as:

$$h_l^t(\theta) \triangleq \sum_{i \in \mathcal{S}^t} w_{i,l}^t f_i(\theta). \tag{10}$$

In essence, the layer-wise gradient of this objective function represents the update of the aggregated global update Δ^t. This relationship is elaborated in the following lemma.

Lemma 4.4. *With Assumption 4.2, we have:*

$$\mathbb{E}[\Delta^t] = \sum_{l \in \mathcal{L}_t} \nabla_l h_l^t(\theta^t), \tag{11}$$

where the expectation is with respect to mini-batch data sampling.

Proof. We rewrite both sides of Equation (11) by using the definitions in Equations (5) and (10), and apply Assumption 4.2 to obtain the result. □

As aforementioned, the underlying loss function $h_l^t(\theta)$ deviates from the desired global loss function $f(\theta)$, which hinders the optimization of the global model and may lead to suboptimal model performance. Such deviation can be quantified by the difference between the underlying update $\sum_{l\in\mathcal{L}_t} \nabla_l h_l^t(\theta^t)$ and the global gradient $\nabla f(\theta^t)$, i.e.,

$$\mathcal{E}_t \triangleq \left\| \nabla f(\theta^t) - \sum_{l\in\mathcal{L}_t} \nabla_l h_l^t(\theta^t) \right\|^2. \tag{12}$$

The term $\mathcal{E}_t$ can be further decomposed using the Jensen's inequality into two parts:

$$\mathcal{E}_t \leq 2 \left\| \nabla f(\theta^t) - \sum_{l\in\mathcal{L}_t} \nabla_l f(\theta^t) \right\|_2^2 + 2 \left\| \sum_{l\in\mathcal{L}_t} \nabla_l f(\theta^t) - \nabla_l h_l^t(\theta^t) \right\|_2^2. \tag{13}$$

Remark 4.5. These two terms in the right-hand side (RHS) of (13) can be interpreted as follows: (i) The first term is the difference between the gradient w.r.t. all layers and the gradient w.r.t. the selected layers. The value of this term becomes smaller when the selected layers have *larger gradients*; (ii) The second term represents the mismatch between the desired gradient computed by all clients (i.e., $\nabla_l f(\theta^t) = \sum_{i\in\mathcal{N}} \alpha_i \nabla_l f_i(\theta^t)$) and the underlying update computed by partial clients (i.e., $\nabla_l h_l^t(\theta^t) = \sum_{i\in\mathcal{S}_t} w_{i,l}^t \nabla_l f_i(\theta^t)$), resulting from *different layer choices* among clients. If some layer is selected by all clients, its corresponding term in this term can be diminished.

For a better understanding, the following lemma shows an upper bound for the value of $\mathcal{E}_t$.

Lemma 4.6. *With Assumption 4.3, we have:*

$$\mathcal{E}_t \leq 2 \underbrace{\left[\left\| \sum_{l\notin\mathcal{L}_t} \nabla f(\theta^t) \right\|^2 \right]}_{\mathcal{E}_{t,1}} + 2 \underbrace{\sum_{l\in\mathcal{L}_t} \chi_{\mathbf{w}_{t,l}\|\alpha} \kappa_l^2}_{\mathcal{E}_{t,2}}, \tag{14}$$

where $\chi_{\mathbf{w}_{t,l}\|\alpha} \triangleq \sum_{i\in\mathcal{N}} \frac{(w_{i,l}^t - \alpha_i)^2}{\alpha_i}$.

The proofs can be found in Appendix A.1.

Next we aim to analyze the impact of layer selection on the convergence of the global model. Following previous studies (Bottou et al., 2018; Wang et al., 2020), we consider an algorithm to have achieved convergence if it converges to a stationary point of the global loss function, namely, if its expected squared gradient norm $\min_{t\in[T]} \mathbb{E}\left[\|\nabla f(\theta^t)\|_2^2\right]$ is zero. The following theorem and corollary show the convergence of the proposed selective layer fine-tuning framework for FL.

Theorem 4.7. *Define a constant* $C \triangleq 1 - 4\eta L > 0$. *With Assumptions 4.1-4.3, we have:*

$$\min_{t\in[T]} \mathbb{E}\left[\left\|\nabla f(\theta^t)\right\|_2^2\right] \leq \frac{2}{\eta C T}\left[f(\theta^0) - f(\theta^*)\right] + \frac{2\gamma\eta}{C}\sigma^2 + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{\gamma\eta C} + 2\right)(\mathcal{E}_{t,1} + \mathcal{E}_{t,2}), \tag{15}$$

where $\theta^* = \arg\min_{\theta\in\mathbb{R}^P} f(\theta)$ *is the best model with the minimal loss.*

The proofs can be found in Appendix A.2.

Corollary 4.8. *With the commonly selected learning rate* $\eta = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$, *the RHS of (15) except the last term becomes zero as* $T \to \infty$. *Therefore, FL with selective layer fine-tuning may only oscillate around a stationary point of the global loss function with a non-zero error floor* $\mathcal{O}(\mathcal{E}_{t,1} + \mathcal{E}_{t,2})$.

According to Theorem 4.7 and Corollary 4.8, the training performance of the global model is degraded by the increase of $\mathcal{E}_{t,1} + \mathcal{E}_{t,2}$, in the following aspects:

- The term $\mathcal{E}_{t,1}$ indicates that it might lead to a suboptimal global model if layers with large gradient norms were not selected for fine-tuning.
- The term $\mathcal{E}_{t,2}$ shows that the consistent selections among clients promote the convergence of the global model. Specifically, if the l-th layer has large gradient diversity κ_l, implying significant objective bias among clients, reducing the weight divergence $\chi_{\mathbf{w}_{t,l}\|\alpha}$ helps to alleviate this term.

These findings highlight that the layer selection strategy that minimizes both terms can achieve better convergence of the global model. However, minimizing these two terms simultaneously may lead to contradictory selection decisions. Moreover, the optimal solution for minimizing the sum of $\mathcal{E}_{t,1} + \mathcal{E}_{t,2}$ is inaccessible, since the ground-truth values are intractable in practice. To resolve these challenges, in the next subsection, we propose a strategy to adaptively select layers for clients and promote the learning performance of the global model.

4.2 Layer Selection Strategy

Based on the above analysis, we need to determine the selected layer sets $\{\mathcal{L}_t^i\}$ that minimize the values of $\mathcal{E}_{t,1}$ and $\mathcal{E}_{t,2}$. As both terms are hard to compute directly, we first design an approach to estimate their values.

To minimize $\mathcal{E}_{t,1}$, we prefer selecting layers with larger gradients, which can be achieved by maximizing the value of $\sum_{l\in\mathcal{L}_t} \|\nabla_l f(\theta^t)\|_2^2$. Since the norm of the global gradient $\nabla_l f(\theta^t)$ is unknown, we estimate it by using the sum of stochastic local gradients, expressed as $\sum_{i\in\mathcal{S}^t} \|g_{i,l}(\theta^t;\xi_i^t)\|_2^2$. Meanwhile, forcing the same layer selection among clients can reduce the value of $\chi_{\mathbf{w}_{t,l}\|\alpha}$ and thus alleviate the term $\mathcal{E}_{t,2}$. For this purpose, we introduce the regularization term $\sum_{j\neq i} \|\mathbf{m}_i^t - \mathbf{m}_j^t\|_1$ into the optimization objective. Therefore, the selection of layers is determined by solving the following optimization problem:

$$\text{(P1)} \quad \begin{aligned} &\max_{\{\mathbf{m}_i^t\}} \sum_{i\in\mathcal{S}^t}\sum_{l\in\mathcal{L}_i^t} \|g_{i,l}(\theta^t;\xi_i^t)\|_2^2 - \frac{\lambda}{2}\sum_{i\in\mathcal{S}^t}\sum_{j\neq i} \|\mathbf{m}_i^t - \mathbf{m}_j^t\|_1^2, \\ &\text{s.t. } \mathcal{R}(\mathbf{m}_i^t) \leq R_i^t, \quad \forall i \in \mathcal{S}^t. \end{aligned}$$

Here $\lambda \geq 0$ is a weighting constant. Specifically, a large λ forces consistent selection across clients, while $\lambda = 0$ allows for independent choices among clients. The constraint in Problem (P1) ensures that the total cost of the selected layers meets the clients' local resource budgets R_i^t, and the cost function $\mathcal{R}(\cdot)$ is typically a linear function of $\mathbf{m}_i^t$.

Solving Problem (P1) further gives an effective layer selection strategy for clients. At the beginning of a training epoch, each participating client $i \in \mathcal{S}^t$ evaluates the current global model θ^t on a batch of local data and obtains the layer-wise gradient $g_{i,l}(\theta^t;\xi_i^t), \forall l \in \mathcal{L}$. Subsequently, clients upload the norms of these gradients $\|g_{i,l}(\theta^t;\xi_i^t)\|_2, \forall l \in \mathcal{L}$, which are L-dimensional vectors, to the server. With these values, the server can optimize the selected layer sets for clients by solving Problem (P1).

In general, the proposed layer selection strategy leads to client-specific layer sets, determined based on the estimated gradient norms. Meanwhile, a hyper-parameter λ is used to regulate the extent to which clients are encouraged to select the same layer. In the next section, we empirically demonstrate the benefits of the proposed strategy in effectively identifying critical layers and achieving better model performance than existing methods.

4.3 Discussions on Computational and Communication costs

In this section, we provide discussions on the computational and communication costs, considering a case where each client selects R layers to fine-tune a model with a total of L layers.

Computational costs Since both the proposed method and full model fine-tuning require the same forward operations, we focus on comparing the computational costs of backward operations among different methods. For simplicity, we assume each layer requires b FLOPs of backward operations. The average computational costs of the proposed layer selection method are calculated as:

$$\text{Cost}_{\text{ours}} = \underbrace{b(L-1)}_{\text{Select}} + \underbrace{bR\tau}_{\text{Fine-tune}} = b(R\tau + L - 1), \tag{16}$$

where τ represents the local training steps. For comparison, fully fine-tuning a model requires the computational costs of:

$$\text{Cost}_{\text{full}} = bL\tau = \frac{L\tau}{R\tau + L - 1}\text{Cost}_{\text{ours}}. \tag{17}$$

As a result, the proposed method takes a much lower computational cost than full model fine-tuning, and the cost reduction is proportional to the number of layers and local training steps.

Meanwhile, the layer selection step in the proposed method introduces slightly additional costs of $\frac{L-R}{\tau L}\text{Cost}_{\text{full}}$, which can be further reduced by evaluating the model on a smaller volume of data or making the selection decision at a lower frequency.

Communication costs The communication costs are determined by the transmitted bits during the training process. The proposed method only needs to transmit the selected layers, whose communication costs are much lower than those of full model fine-tuning that needs to upload the entire model. For example, assuming that different layers have the same number of parameters, the communication cost of the proposed method is $\frac{R}{L}$ of full model fine-tuning.

To summarize, the computational and communication costs of the proposed method are much lower than those of full model fine-tuning. More empirical evidence can be found in Section 5.3.

5 EXPERIMENTS

5.1 SETTINGS

Datasets & Models We conduct a series of experiments on several widely-used image classification datasets, including CIFAR-10 (Krizhevsky & Hinton, 2009) and DomainNet (Peng et al., 2019), text classification dataset, i.e., XGLUE-NC (Liang et al., 2020), and five benchmark question-answering (QA) datasets, including SCIQ (Welbl et al., 2017), OpenbookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), ARC-Easy and ARC-Challenge (Bhakthavatsalam et al., 2021) datasets. More details of the adopted datasets can be found in in Appendix B.1.

As for the splitting of datasets, inspired by previous studies (Zhu et al., 2021; Kim et al., 2023), we consider two commonly observed data heterogeneity among clients: (i) *Label skew*: adopting Dirichlet distribution to allocate data samples of the CIFAR-10 dataset; (ii) *Feature skew*: adopting naturally domain shift on the DomainNet, XGLUE-NC, and QA datasets. Specifically, each QA dataset is equally divided into two subsets, with each client possessing one subset of samples from one of the five datasets. We adopt the CLIP model (Radford et al., 2021) for image classification tasks and the multi-lingual XLM-Roberta-Base (Conneau et al., 2019) model on the XGLUE-NC dataset. In addition, we train a LLaMA-2-7B (Touvron et al., 2023) model on the QA dataset.

Server & Clients In the experiments, we set up an FL system with a central server and $N = 100$ clients. In each training epoch, the server randomly selects a subset of 20 clients, and broadcasts the up-to-date model to these clients for local training. Besides, there are $N = 10$ clients in the QA task and five clients are randomly chosen for training in each epoch. The resource budgets of clients are limited, which are quantified as the maximum number of layers they can fine-tune in the local training. For example, we use $R_i = 1$ to indicate that the resource of client i cannot afford fine-tuning more than 1 layer. The resource budgets can be identical or heterogeneous among clients.

Baselines We compare the proposed layer selection strategy with several competitive baselines, including: (i) **Top** (Kovaleva et al., 2019; Lee et al., 2019b): Clients only fine-tune the top few layers (near the output) based on their task-specific data; (ii) **Bottom** (Lee et al., 2022): Clients only fine-tune the bottom few layers (near the input) based on their task-specific data, which can be beneficial for the tasks with input-shift; (iii) **Both** (Xiao et al., 2023): Clients fine-tune an equal proportion of both the top and bottom layers, which shows the effectiveness for large language models; (iv) **SNR** (Mahsereci et al., 2017): Clients fine-tune the layers with higher signal-to-noise ratio (SNR) values, defined as the ratio of the mean of gradient elements to their variance; (v) **RGN** (Cheng et al., 2023; Lee et al., 2022): Clients fine-tune the layers with higher relative gradient norm (RGN) values, defined as the ratio of gradient norm to the parameter norm; (vi) Moreover, we consider **Full** model fine-tuning, i.e., training the entire model, as the performance benchmark. More implementation details can be found in Appendix B.2.

5.2 COMPARISONS

We conduct experiments with the *identical resource* scenario and the *heterogeneous resource* scenario.

Identical resource scenario We first consider clients with identical computational resources, i.e., clients select the same number of layers ($R_i = R, \forall i \in \mathcal{N}$) for fine-tuning. The experimental results are shown in Table 1. From the model performance (accuracy) on the CIFAR-10 and DomainNet

Table 1: Test accuracy (%) on both image and text datasets, where each client selects R layers for fine-tuning. The best results are highlighted in **bold**.

	CIFAR-10		DomainNet		XGLUE-NC		QA	
	$R=1$	$R=2$	$R=1$	$R=2$	$R=1$	$R=2$	$R=1$	$R=2$
Full	95.43		90.27		82.11		65.98	
Top	93.09	93.61	87.86	88.32	69.86	77.05	63.90	64.44
Bottom	27.38	32.81	13.80	18.63	40.43	40.60	64.18	64.60
Both	-	94.96	-	85.48	-	74.65	-	64.41
SNR	94.47	90.49	86.38	87.67	69.11	79.92	63.80	64.58
RGN	92.69	89.48	88.80	87.19	74.06	79.48	63.73	64.70
Ours	**95.47**	**96.05**	**89.37**	**89.64**	**74.95**	**80.39**	**64.71**	**65.03**

Table 2: Test accuracy (%) on both image and text datasets, where clients have different resources ($R_i \in [1, 4]$). The best results are highlighted in **bold**.

	CIFAR-10	DomainNet	XGLUE-NC	QA
Full	95.43	90.27	82.11	65.98
Top	91.22	89.29	78.17	64.10
Bottom	27.38	23.10	50.92	64.51
Both	89.91	86.27	73.01	64.64
SNR	75.72	87.34	78.24	64.51
RGN	93.83	88.19	79.36	64.56
Ours	**95.57**	**89.39**	**80.18**	**65.80**

datasets, we observe that the proposed strategy demonstrates notable superiority over partial layer fine-tuning baselines. Specifically, fine-tuning only one layer of the CLIP model achieves comparable performance with tuning the entire model, since the CLIP model is sufficiently powerful to extract useful features and thus requires less training on task-specific data. This also reveals that selective layer fine-tuning well meets the performance requirement within the resources of clients.

On text datasets, including XGLUE-NC and QA, the proposed layer selection strategy and RGN demonstrate similar performance, both surpassing other baseline methods (especially Top and Both) by noticeable margins. One potential explanation for this phenomenon could be that they result in similar layer selections, indicating that updating layers with higher relative gradient norms is more beneficial than other strategies, which is consistent with previous study (Lee et al., 2022). Moving a forward step, the proposed method adopts a flexible and dynamic layer selection strategy instead of fixed strategies, which leads to competitive performance.

Heterogeneous resource scenario Further, we conduct experiments with heterogeneous clients, i.e., clients have different local resources and thus tend to select different numbers of layers for fine-tuning. Such a heterogeneous resource scenario is more practical (Yang et al., 2021; Chai et al., 2019) and brings additional challenges for selective layer fine-tuning. Inspired by previous studies (Wang et al., 2020; Nguyen et al., 2022b), the number of layers to be fine-tuned, denoted as R_i for client i, is sampled from a truncated half Normal distribution within $[1, 4]$.

The experimental results are shown in Table 2, from which we observe that the proposed strategy consistently shows superiority over all the baseline methods on all the datasets. Compared with baselines, the proposed strategy allows clients to flexibly determine the proper number of layers to be tuned and effectively find the most important layers. This advantage arises from enabling clients with sufficient resources to prioritize the selection of more critical layers instead of being restricted to layers in fixed positions. Overall, these experimental results demonstrate the advantage of the proposed strategy when handling heterogeneity in real-world FL applications.

5.3 Further Discussions

Visualization of selected layers For a better understanding on the proposed layer selection strategy,

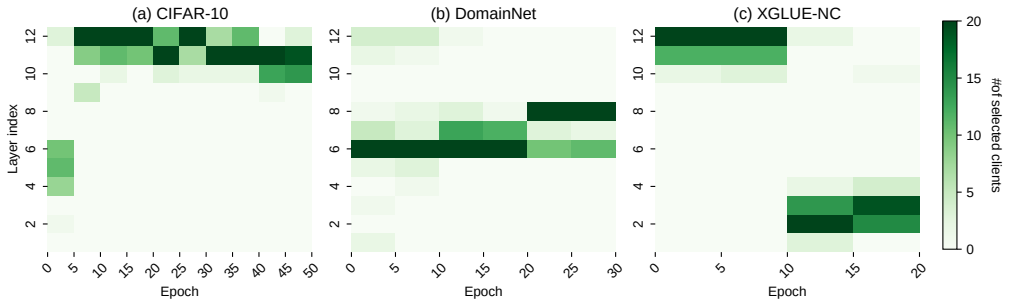

Figure 2: Visualization of selected layers ($R_i \in [1, 4]$).

Table 3: Comparisons of computational and communication costs when fine-tuning the CLIP model on the CIFAR-10 ($R = 1$). The numbers in brackets represent the costs of the proposed layer selection strategy.

	Computational cost (TFLOPs)	Ratio	Transmission (MBits)	Ratio
Full Model Fine-tuning	8.47	100%	2,811	100%
Proposed Method	2.24 (1.51)	26% (17%)	234	8.33%
Proposed (Sel. Period=2)	1.46 (0.75)	17% (9.5%)	234	8.33%
Proposed (Sel. Batch=1)	0.99 (0.30)	12% (3.4%)	234	8.33%

we visualize the selected layers on different datasets in Figure 2. When fine-tuning the CLIP model on the CIFAR-10 dataset, it can be observed that the focus is primarily on updating a few top layers, indicating that the low-level features (related to middle and bottom layers) are transferable from pre-trained data to downstream tasks. In comparison, the DomainNet dataset, characterized by a significant domain shift, necessitates extensive tuning of the middle layers in the CLIP model. Furthermore, on the XGLUE-NC dataset, we observe a clear progression of selected layers for fine-tuning, with a shift from the top layers progressively down to the bottom layers. Such a pattern is markedly different from the trend observed in the image datasets. One possible reason lies in the intrinsic differences between the modalities of text and image data. These results highlight the necessity for adaptive layer selection and adjustment strategies in FL to accommodate varying dataset properties and domain shifts.

Comparisons regarding computational and communication costs We compare the computational costs (in TFLOPs) and communication costs (in transmitted MBits) of the proposed method with full model fine-tuning when adopting the CLIP model on the CIFAR-10. For the proposed method, we consider fine-tuning only one layer, as it is sufficient to achieve comparable accuracy with full model fine-tuning according to Table 1. The results in Table 3 evidence a substantial decrease in both computational and communication requirements when utilizing the proposed method. Besides, we can observe that the layer selection strategy takes as low as 3.4% of the computational costs. These experimental results demonstrate that the proposed method significantly reduces both the computational and communication costs compared to full model fine-tuning.

6 Conclusions

In this paper, we study a practical FL setting for fine-tuning foundation models, where clients are allowed to optimize a subset of layers using their task-specific data. We carefully consider the impact of both data heterogeneity and device heterogeneity across clients, providing a comprehensive theoretical analysis of the optimization objective of selective layer fine-tuning and global model convergence. The theoretical analysis offers insights into how the selected layers influence global model training and highlights the role of layer importance and client heterogeneity. We further propose a novel strategy for layer selection that considers the local data and available resources at clients. The experimental results demonstrate that the proposed strategy outperforms baseline strategies in improving the global model training performance and even matches full model fine-tuning performance in some scenarios, showing the potential for more efficient and tailored real-world FL applications of the proposed layer selection strategy.

Reproducibility statement The assumptions and proofs of theoretical results in this work are given in Section 4.1 and Appendix A.1. The experimental settings are described in Section 5.1 and Appendix B. The source codes are available at `https://anonymous.4open.science/r/fed_selected_tune/`.

REFERENCES

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. Think you have solved direct-answer question answering? Try ARC-DA, the direct-answer AI2 reasoning challenge. *arXiv preprint arXiv:2102.03315*, 2021.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pp. 7432–7439, 2020.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Keith Bonawitz et al. Towards federated learning at scale: System design. In *Proceedings of Machine Learning and Systems*, 2019.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, Aug. 2018.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems (NeurIPS)*, pp. 1877–1901, 2020.

Zheng Chai, Hannan Fayyaz, Zeshan Fayyaz, Ali Anwar, Yi Zhou, Nathalie Baracaldo, Heiko Ludwig, and Yue Cheng. Towards taming the resource and data heterogeneity in federated learning. In *2019 USENIX conference on operational machine learning (OpML 19)*, pp. 19–21, 2019.

Chaochao Chen, Xiaohua Feng, Jun Zhou, Jianwei Yin, and Xiaolin Zheng. Federated large language model: A position paper. *arXiv preprint arXiv:2307.08925*, 2023a.

Daoyuan Chen, Liuyi Yao, Dawei Gao, Bolin Ding, and Yaliang Li. Efficient personalized federated learning via sparse model-adaptation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 5234–5256, 2023b.

Jinyu Chen, Wenchao Xu, Song Guo, Junxiao Wang, Jie Zhang, and Haozhao Wang. FedTune: A deep dive into efficient federated fine-tuning with pre-trained transformers. *arXiv preprint arXiv:2211.08025*, 2022.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations. *arXiv preprint arXiv:2308.06767*, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations (ICLR)*, 2021.

Chen Dun, Cameron R Wolfe, Christopher M Jermaine, and Anastasios Kyrillidis. Resist: Layer-wise decomposition of resnets for distributed training. In *Uncertainty in Artificial Intelligence (UAI)*, pp. 610–620, 2022.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. GPTs are GPTs: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*, 2023.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. CLIP-Adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021.

Agrin Hilmkil, Sebastian Callh, Matteo Barbieri, Leon René Sütfeld, Edvin Listo Zec, and Olof Mogren. Scaling federated learning for fine-tuning of large language models. In *International Conference on Applications of Natural Language to Information Systems*, pp. 15–23, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 2790–2799, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022.

Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, Jian Li, and M. Hadi Amini. A survey on federated learning for resource-constrained IoT devices. *IEEE Internet of Things Journal*, 9(1):1–24, 2022.

Gal Kaplun, Andrey Gurevich, Tal Swisa, Mazor David, Shai Shalev-Shwartz, and Eran Malach. Less is more: Selective layer finetuning with subtuning. *arXiv preprint arXiv:2302.06354*, 2023.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International conference on machine learning (ICML)*, pp. 5132–5143, 2020.

Yeachan Kim, Junho Kim, Wing-Lam Mok, Jun-Hyung Park, and SangKeun Lee. Client-customized adaptation for parameter-efficient federated learning. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics*, pp. 1159–1172, 2023.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*, 2019.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. [Online]. Available: `https://www.cs.toronto.edu/~kriz/cifar.html`, 2009.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. *arXiv preprint arXiv:2309.00363*, 2023.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models. In *International Conference on Learning Representations (ICLR)*, 2019a.

Jaejun Lee, Raphael Tang, and Jimmy Lin. What would elsa do? Freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*, 2019b.

Sunwoo Lee, Tuo Zhang, and A Salman Avestimehr. Layer-wise adaptive model aggregation for scalable federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 8491–8499, 2023.

Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. In *International Conference on Learning Representations (ICLR)*, 2022.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *38th IEEE International Conference on Data Engineering (ICDE)*, pp. 965–978, 2022.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4582–4597, 2021.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-iid features via local batch normalization. In *9th International Conference on Learning Representations (ICLR)*, 2021.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6008–6018, 2020.

Maren Mahsereci, Lukas Balles, Christoph Lassner, and Philipp Hennig. Early stopping without a validation set. *arXiv preprint arXiv:1703.09580*, 2017.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2381–2391, 2018.

John Nguyen, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. Where to begin? Exploring the impact of pre-training and initialization in federated learning. *arXiv preprint arXiv:2206.15387*, 2022a.

John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. Federated learning with buffered asynchronous aggregation. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 3581–3607, 2022b.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1406–1415, 2019.

Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 17716–17758. PMLR, 2022.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8821–8831, 2021.

Zhiqiang Shen, Zechun Liu, Jie Qin, Marios Savvides, and Kwang-Ting Cheng. Partial is better than all: Revisiting fine-tuning strategy for few-shot learning. *arXiv preprint arXiv:2102.03983*, 2021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin (eds.), *Proceedings of the 3rd Workshop on Noisy User-generated Text (NUT@EMNLP)*, pp. 94–106, 2017.

Guangxuan Xiao, Ji Lin, and Song Han. Offsite-tuning: Transfer learning without full model. *arXiv preprint arXiv:2302.04870*, 2023.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. Raise a child in large language model: Towards effective and generalizable fine-tuning. *arXiv preprint arXiv:2109.05687*, 2021.

Chengxu Yang, Qipeng Wang, Mengwei Xu, Zhenpeng Chen, Kaigui Bian, Yunxin Liu, and Xuanzhe Liu. Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data. In *Proceedings of the Web Conference 2021*, pp. 935–946, 2021.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Haojie Zhang, Ge Li, Jia Li, Zhongjin Zhang, Yuqi Zhu, and Zhi Jin. Fine-tuning pre-trained language models effectively by optimizing subnetworks adaptively. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 21442–21454, 2022a.

Kaiyan Zhang, Ning Ding, Biqing Qi, Xuekai Zhu, Xinwei Long, and Bowen Zhou. CRaSh: Clustering, removing, and sharing enhance fine-tuning without full large language model. *arXiv preprint arXiv:2310.15477*, 2023.

Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10174–10183, 2022b.

Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-IID data: A survey. *Neurocomputing*, 465:371–390, 2021.

A CONVERGENCE ANALYSIS: FULL PROOFS

A.1 USEFUL LEMMAS

A.1.1 ONE-ROUND LOSS DECAY

Lemma A.1. *With Assumption 4.1, we have:*

$$\mathbb{E}[f(\theta^{t+1})] - \mathbb{E}[f(\theta^t)] \leq \frac{1}{2\gamma}\mathcal{E}_t + \underbrace{\mathbb{E}\left\langle \sum_{l\in\mathcal{L}_t} \nabla_l h_l^t(\theta^t), \theta^{t+1} - \theta^t \right\rangle}_{\mathcal{T}_1} + \gamma \underbrace{\mathbb{E}\left[\left\|\theta^{t+1} - \theta^t\right\|^2\right]}_{\mathcal{T}_2}. \tag{18}$$

Proof. We begin with analyzing the loss decay by using γ-smoothness in Assumption 4.1 as follows:

$$\mathbb{E}[f(\theta^{t+1})] - \mathbb{E}[f(\theta^t)] \tag{19}$$

$$\leq \mathbb{E}\langle \nabla f(\theta^t), \theta^{t+1} - \theta^t \rangle + \frac{\gamma}{2}\mathbb{E}[\left\|\theta^{t+1} - \theta^t\right\|^2] \tag{20}$$

$$= \mathbb{E}\left\langle \nabla f(\theta^t) - \sum_{l\in\mathcal{L}_t} \nabla_l h_l^t(\theta^t) + \sum_{l\in\mathcal{L}_t} \nabla_l h_l^t(\theta^t), \theta^{t+1} - \theta^t \right\rangle + \frac{\gamma}{2}\mathbb{E}[\left\|\theta^{t+1} - \theta^t\right\|^2] \tag{21}$$

$$= \underbrace{\mathbb{E}\left\langle \nabla f(\theta^t) - \sum_{l\in\mathcal{L}_t} \nabla_l h_l^t(\theta^t), \theta^{t+1} - \theta^t \right\rangle}_{\mathcal{T}_0} + \underbrace{\mathbb{E}\left\langle \sum_{l\in\mathcal{L}_t} \nabla_l h_l^t(\theta^t), \theta^{t+1} - \theta^t \right\rangle}_{\mathcal{T}_1} + \frac{\gamma}{2}\underbrace{\mathbb{E}\left[\left\|\theta^{t+1} - \theta^t\right\|^2\right]}_{\mathcal{T}_2}. \tag{22}$$

By Young's inequality, we upper bound the term $\mathcal{T}_0$ as:

$$\mathcal{T}_0 = \mathbb{E}\left\langle \nabla f(\theta^t) - \sum_{l\in\mathcal{L}_t} \nabla_l h_l^t(\theta^t), \theta^{t+1} - \theta^t \right\rangle \tag{23}$$

$$\leq \frac{1}{2\gamma}\underbrace{\mathbb{E}\left[\left\|\nabla f(\theta^t) - \sum_{l\in\mathcal{L}_t} \nabla_l h_l^t(\theta^t)\right\|^2\right]}_{\mathcal{E}_t} + \frac{\gamma}{2}\mathbb{E}\left[\left\|\theta^{t+1} - \theta^t\right\|^2\right] \tag{24}$$

$$= \frac{1}{2\gamma}\mathcal{E}_t + \frac{\gamma}{2}\mathcal{T}_2. \tag{25}$$

Plugging (25) back into (22) gives the result in (18).

□

A.1.2 ANALYZING $\mathcal{E}_t$: PROOF OF LEMMA 4.6

In this subsection, we prove the result in Lemma 4.6.

We begin with decomposing the term $\mathcal{E}_t$ using the Jensen's inequality as:

$$\mathcal{E}_t \leq 2\underbrace{\|\nabla f(\theta^t) - \sum_{l\in\mathcal{L}_t} \nabla_l f(\theta^t)\|_2^2}_{\tilde{\mathcal{E}}_{t,1}} + 2\underbrace{\|\sum_{l\in\mathcal{L}_t} \nabla_l f(\theta^t) - \nabla_l h_l^t(\theta^t)\|_2^2}_{\tilde{\mathcal{E}}_{t,2}}. \tag{26}$$

For the first term $\tilde{\mathcal{E}}_{1,t}$, we directly obtain:

$$\tilde{\mathcal{E}}_{1,t} = \mathbb{E}\left[\left\|\nabla f(\theta^t) - \sum_{l\in\mathcal{L}_t} \nabla_l f(\theta^t)\right\|^2\right] = \mathbb{E}\left[\left\|\sum_{l\notin\mathcal{L}_t} \nabla_l f(\theta^t)\right\|^2\right]. \tag{27}$$

Afterwards, we derive the value of $\tilde{\mathcal{E}}_{2,t}$ as follows:

$$\tilde{\mathcal{E}}_{2,t} =\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l f(\theta^t)-\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right] \tag{28}$$

$$=\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}\alpha_i\nabla_l f_i(\theta^t)-\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{S}_t}w_{i,l}^t\nabla_l f_i(\theta^t)\right\|^2\right] \tag{29}$$

$$=\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}\alpha_i\nabla_l f_i(\theta^t)-\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta^t)\right\|^2\right] \tag{30}$$

$$=\sum_{l\in\mathcal{L}_t}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{N}}\frac{w_{i,l}^t-\alpha_i}{\sqrt{\alpha_i}}\sqrt{\alpha_i}\left(\nabla_l f_i(\theta^t)-\nabla_l f(\theta^t)\right)\right\|^2\right] \tag{31}$$

$$\leq\sum_{l\in\mathcal{L}_t}\left[\sum_{i\in\mathcal{N}}\frac{(w_{i,l}^t-\alpha_i)^2}{\alpha_i}\right]\left[\sum_{i\in\mathcal{N}}\alpha_i\mathbb{E}\left[\left\|\nabla_l f_i(\theta^t)-\nabla_l f(\theta^t)\right\|^2\right]\right] \tag{32}$$

$$\leq\sum_{l\in\mathcal{L}_t}\chi_{\mathbf{w}_{t,l}\|\alpha}\kappa_l^2. \tag{33}$$

where (32) follows the Cauchy–Schwarz inequality and (33) applies Assumption 4.3.

By substituting the RHS of (27) and (33) into (26), we complete the proof. □

A.2 Convergence Analysis for Single-Step Case

In this subsection, we consider $\tau = 1$ and prove Theorem 4.7.

We derive the value of $\mathcal{T}_1$ as follows:

$$\mathcal{T}_1=\mathbb{E}\left\langle\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t),-\eta\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\rangle=-\eta\mathbb{E}\left[\left\|\sum_{l\in L_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]. \tag{34}$$

Afterwards, we give an upper bound for the term $\mathcal{T}_2$ as follows:

$$\mathcal{T}_2=\left[\left\|\eta\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{S}^t}w_{i,l}^t g_{i,l}(\theta^t;\xi_i^t)\right\|^2\right] \tag{35}$$

$$\leq\eta^2\mathbb{E}\left[\left\|\sum_{l\in L_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+\eta^2\sigma^2, \tag{36}$$

where (36) follows Assumption 4.2.

Using the result in Lemma A.1, we have:

$$\begin{aligned}&\mathbb{E}[f(\theta^{t+1})]-\mathbb{E}[f(\theta^t)]\\ \leq&\frac{1}{2\gamma}\mathcal{E}_t-\eta\mathbb{E}\left[\left\|\sum_{l\in L_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+\gamma\left[\eta^2\mathbb{E}\left[\left\|\sum_{l\in L_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+\eta^2\sigma^2\right] &(37)\\ =&\frac{1}{2\gamma}\mathcal{E}_t-\eta(1-\gamma\eta)\mathbb{E}\left[\left\|\sum_{l\in L_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+\gamma\eta^2\sigma^2. &(38)\end{aligned}$$

We define a constant $C\triangleq 1-\gamma\eta>0$ and arrange the terms in (38) as follows:

$$\mathbb{E}\left[\left\|\sum_{l\in L_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]\leq\frac{1}{\eta C}\left[\mathbb{E}[f(\theta^t)]-\mathbb{E}[f(\theta^{t+1})]\right]+\frac{1}{2\gamma\eta C}\mathcal{E}_t+\frac{\gamma\eta}{C}\sigma^2. \tag{39}$$

By Jensen's inequality, we have:

$$\mathbb{E}\left[\left\|\nabla f(\theta^t)\right\|^2\right] \tag{40}$$

$$=\mathbb{E}\left[\left\|\nabla f(\theta^t) - \sum_{l\in L_t}\nabla_l h_l^t(\theta^t) + \sum_{l\in L_t}\nabla_l h_l^t(\theta^t)\right\|^2\right] \tag{41}$$

$$\leq 2\mathbb{E}\left[\left\|\nabla f(\theta^t) - \sum_{l\in L_t}\nabla_l h_l^t(\theta^t)\right\|^2\right] + 2\mathbb{E}\left[\left\|\sum_{l\in L_t}\nabla_l h_l^t(\theta^t)\right\|^2\right] \tag{42}$$

$$=2\mathcal{E}_t + 2\mathbb{E}\left[\left\|\sum_{l\in L_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]. \tag{43}$$

Combining (39) and (43) gives:

$$\mathbb{E}\left[\left\|\nabla f(\theta^t)\right\|^2\right] \leq \frac{2}{\eta C}\left[\mathbb{E}[f(\theta^t)] - \mathbb{E}[f(\theta^{t+1})]\right] + \left(\frac{1}{\gamma\eta C} + 2\right)\mathcal{E}_t + \frac{2\gamma\eta}{C}\sigma^2. \tag{44}$$

We sum up both sides of (44) over $t = 0, 1, \ldots, T-1$ and divide them by T to obtain the following result:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla f(\theta^t)\right\|^2\right]$$

$$\leq\frac{2}{\eta CT}\left[\mathbb{E}[f(\theta^0)] - \mathbb{E}[f(\theta^T)]\right] + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{\gamma\eta C} + 2\right)\mathcal{E}_t + \frac{2\gamma\eta}{C}\sigma^2 \tag{45}$$

$$\leq\frac{2}{\eta CT}\left[f(\theta^0) - f(\theta^*)\right] + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{\gamma\eta C} + 2\right)\mathcal{E}_t + \frac{2\gamma\eta}{C}\sigma^2 \tag{46}$$

$$\leq\frac{2}{\eta CT}\left[f(\theta^0) - f(\theta^*)\right] + \frac{2\gamma\eta}{C}\sigma^2 + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{\gamma\eta C} + 2\right)(\mathcal{E}_{t,1} + \mathcal{E}_{t,2}). \tag{47}$$

□

A.3 Convergence Analysis for Multi-Step Case

Consider the general case where $\tau > 1$. We characterize the convergence in the following theorem and note that the impact of $\mathcal{E}_{t,1} + \mathcal{E}_{t,2}$ is similar to that in Theorem 4.7.

Theorem A.2. *Let* $C' \triangleq 1 - 4\eta\tau - 8\eta^2\gamma^2\tau(\tau-1) - 32\eta^3\gamma^2\tau^2(\tau-1) > 0$ *and* $A_\tau \triangleq \eta + 2\eta^2\gamma^2\tau(\tau-1) + 8\eta^3\gamma^2\tau^2(\tau-1)$. *With Assumptions 4.1-4.3, we have:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla f(\theta^t)\right\|^2\right] \leq\frac{2}{\eta\tau C'T}\left[f(\theta^0) - f(\theta^*)\right] + \frac{4A_\tau}{C'}\sigma^2 + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{\eta\tau\gamma C'} + 2\right)(\mathcal{E}_{t,1} + \mathcal{E}_{t,2}). \tag{48}$$

Proof. In Lemma A.1, the term $\mathcal{T}_1$ is related to client drift caused by multiple local SGD steps, which can be upper bounded as follows:

$$\mathcal{T}_1$$

$$= -\eta\sum_{k=0}^{\tau-1}\mathbb{E}\left\langle\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t), \sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}} w_{i,l}^t\nabla_l f_i(\theta_i^{t,k})\right\rangle \tag{49}$$

$$= -\eta\sum_{k=0}^{\tau-1}\mathbb{E}\left\langle\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t), \sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\rangle$$

$$+\eta\sum_{k=0}^{\tau-1}\mathbb{E}\left\langle\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t),\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)-\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta_i^{t,k})\right\rangle \tag{50}$$

$$\leq-\frac{\eta}{2}\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+\frac{\eta}{2}\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)-\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta_i^{t,k})\right\|^2\right] \tag{51}$$

$$=-\frac{\eta\tau}{2}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+\frac{\eta}{2}\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta^t)-\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta_i^{t,k})\right\|^2\right] \tag{52}$$

$$\leq-\frac{\eta\tau}{2}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+\underbrace{\frac{\eta\gamma^2}{2}\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\left(\theta^t-\theta_i^{t,k}\right)\right\|^2\right]}_{\mathcal{T}_4}, \tag{53}$$

where (51) uses the inequality $\langle\mathbf{a},\mathbf{b}\rangle\leq\frac{\|\mathbf{a}\|^2}{2}+\frac{\|\mathbf{b}\|^2}{2}$, and (53) follows Assumption 4.1.

Then we analyze the term $\mathcal{T}_2$ as follows:

$$\mathcal{T}_2$$

$$\leq\eta^2\mathbb{E}\left[\left\|\sum_{k=0}^{\tau-1}\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta_i^{t,k})\right\|^2\right]+\eta^2\tau\sigma^2 \tag{54}$$

$$\leq\eta^2\tau\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta_i^{t,k})-\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta^t)+\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta^t)\right\|^2\right]$$
$$+\eta^2\tau\sigma^2 \tag{55}$$

$$\leq 2\eta^2\tau\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta_i^{t,k})-\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta^t)\right\|^2\right]$$
$$+2\eta^2\tau\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+\eta^2\tau\sigma^2 \tag{56}$$

$$\leq 2\eta^2\gamma^2\tau\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\left(\theta_i^{t,k}-\theta^t\right)\right\|^2\right]+2\eta^2\tau^2\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+\eta^2\tau\sigma^2 \tag{57}$$

$$=2\eta^2\gamma^2\tau\mathcal{T}_4+2\eta^2\tau^2\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+\eta^2\tau\sigma^2, \tag{58}$$

where (54) follows Assumption 4.2, (55)-(56) apply the Jensen's inequality, and (57) follows Assumption 4.1.

Following Lemma 22 in (Pillutla et al., 2022), $\mathcal{T}_4$ can be upper bounded as:

$$\mathcal{T}_4\leq\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\theta^t-\theta_i^{t,k}\right\|^2\right] \tag{59}$$

$$\leq 8\eta^2\tau^2(\tau-1)\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t\nabla_l f_i(\theta^t)\right\|^2\right]+\sum_{l\in\mathcal{L}_t}\sum_{i\in\mathcal{N}}w_{i,l}^t 4\eta^2\tau^2(\tau-1)\sigma_l^2 \tag{60}$$

$$=8\eta^2\tau^2(\tau-1)\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]+4\eta^2\tau^2(\tau-1)\sigma^2. \tag{61}$$

Plugging (54),(58) and (61) back into (18), we have the following result:

$$\mathbb{E}[f(\theta^{t+1})] - \mathbb{E}[f(\theta^t)] \tag{62}$$

$$\leq \frac{1}{2\gamma}\mathcal{E}_t - \frac{\eta\tau}{2}\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right] + \frac{\eta\gamma^2}{2}\mathcal{T}_4 + 2\eta^2\gamma^2\tau\mathcal{T}_4 + 2\eta^2\tau^2\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right] + \eta^2\tau\sigma^2 \tag{63}$$

$$= \frac{1}{2\gamma}\mathcal{E}_t - \frac{\eta\tau}{2}\left(1-4\eta\tau\right)\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right] + \eta^2\tau\sigma^2 + \left(\frac{\eta\gamma^2}{2} + 2\eta^2\gamma^2\tau\right)\mathcal{T}_4 \tag{64}$$

$$= \frac{1}{2\gamma}\mathcal{E}_t - \frac{\eta\tau}{2}\left(1-4\eta\tau\right)\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right] + \eta^2\tau\sigma^2$$
$$+ \left(\frac{\eta\gamma^2}{2} + 2\eta^2\gamma^2\tau\right)\left\{8\eta^2\tau^2(\tau-1)\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right] + 4\eta^2\tau^2(\tau-1)\sigma^2\right\} \tag{65}$$

$$= \frac{1}{2\gamma}\mathcal{E}_t - \frac{\eta\tau}{2}\left[1-4\eta\tau-8\eta^2\gamma^2\tau(\tau-1)-32\eta^3\gamma^2\tau^2(\tau-1)\right]\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right]$$
$$+ \left(\eta^2\tau + 2\eta^3\gamma^2\tau^2(\tau-1) + 8\eta^4\gamma^2\tau^3(\tau-1)\right)\sigma^2. \tag{66}$$

Let $C' \triangleq 1-4\eta\tau-8\eta^2\gamma^2\tau(\tau-1)-32\eta^3\gamma^2\tau^2(\tau-1) > 0$ and $A_\tau \triangleq \eta + 2\eta^2\gamma^2\tau(\tau-1) + 8\eta^3\gamma^2\tau^2(\tau-1)$. We have:

$$\mathbb{E}\left[\left\|\sum_{l\in\mathcal{L}_t}\nabla_l h_l^t(\theta^t)\right\|^2\right] \leq \frac{2}{\eta\tau C'}\left[\mathbb{E}[f(\theta^t)] - \mathbb{E}[f(\theta^{t+1})]\right] + \frac{1}{\eta\tau\gamma C'}\mathcal{E}_t + \frac{2}{C'}A_\tau\sigma^2. \tag{67}$$

Using the result in (43), we have:

$$\mathbb{E}\left[\left\|\nabla f(\theta^t)\right\|^2\right] \leq \frac{4}{\eta\tau C'}\left[\mathbb{E}[f(\theta^t)] - \mathbb{E}[f(\theta^{t+1})]\right] + \left(\frac{1}{\eta\tau\gamma C'} + 2\right)\mathcal{E}_t + \frac{4A_\tau}{C'}\sigma^2. \tag{68}$$

We sum up both sides of (68) over $t = 0, 1, \ldots, T-1$ and divide them by T to obtain the following result:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla f(\theta^t)\right\|^2\right]$$

$$\leq \frac{2}{\eta\tau C'T}\left[\mathbb{E}[f(\theta^0)] - \mathbb{E}[f(\theta^T)]\right] + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{\eta\tau\gamma C'} + 2\right)\mathcal{E}_t + \frac{4A_\tau}{C'}\sigma^2 \tag{69}$$

$$\leq \frac{2}{\eta\tau C'T}\left[f(\theta^0) - f(\theta^*)\right] + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{\eta\tau\gamma C'} + 2\right)\mathcal{E}_t + \frac{4A_\tau}{C'}\sigma^2 \tag{70}$$

$$\leq \frac{2}{\eta\tau C'T}\left[f(\theta^0) - f(\theta^*)\right] + \frac{4A_\tau}{C'}\sigma^2 + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{\eta\tau\gamma C'} + 2\right)(\mathcal{E}_{t,1} + \mathcal{E}_{t,2}). \tag{71}$$

□

B Experimental Details

We implement all methods with PyTorch and run experiments on Nvidia V100 GPUs. For fair comparisons, we adopt the same training epochs and hyper-parameters for all methods.

B.1 TRAINING TASKS

To evaluate the methods in various scenarios with different non-IID patterns, we consider two image classification tasks and two text classification tasks.

Table 4: Summary of datasets.

Dataset	Data Type	Non-IID Type	Partition
CIFAR-10	Image	Label skew	$Dir(0.1)$
DomainNet	Image	Feature skew	Domain
XGLUE-NC	Text	Feature skew	Domain
QA	Text	Feature skew	Domain

The image classification tasks include:

- **CIFAR-10** (Krizhevsky & Hinton, 2009): In this training task, we consider the label-skewed case where $P(y_i)$ is different among clients. Following previous works (Li et al., 2022), we adopt Dirichlet allocation with concentration parameter $\alpha = 0.1$ among clients.
- **DomainNet** (Peng et al., 2019): DomainNet contains six domains of data samples, including clipart, real, sketch, infograph, painting, and quickdraw. In this training task, we consider the varying feature case where $P(x_i|y_i)$ is different among clients. Following (Li et al., 2021), each client is allocated random samples from only one domain.

On both tasks, we fine-tune a CLIP Vision Transformer (CLIP) model (Radford et al., 2021).

Besides, we fine-tune an XLM-Roberta-Base model on the following text dataset:

- **XGLUE-NC** (Liang et al., 2020): This is a news classification task consisting of 10 classes. The news texts comprise five languages (English, Spanish, French, German, and Russian). We allocate one random language to each client, which naturally introduces domain shift among clients.

In addition, we evaluate a LLaMA-2-7B model on the QA datasets.

- **QA**: The QA datasets consist of four commonly used question-answering datasets, i.e., SCIQ (Welbl et al., 2017), OpenbookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), ARC-Easy and ARC-Challenge (Bhakthavatsalam et al., 2021). The datasets are transformed into classification tasks where the model determines the correct answer for each question and corresponding choices. We allocate the samples from one random dataset to each client, indicating the domain shift among clients.

B.2 IMPLEMENTATION DETAILS

The CLIP model is pre-trained on the DataComp dataset and is adapted from `https://github.com/openai/CLIP`; the XLM-Roberta-Base model is adapted from `https://huggingface.co/xlm-roberta-base`; the LLaMA-2-7B model is adapted from `https://huggingface.co/meta-llama/LLaMA-7b-chat-hf`. In all training tasks, we freeze the embedding layers of the model and fix the classifier as commonly selected layers (Lee et al., 2019b). The values of adopted hyperparameters are summarized in Table 5. For the proposed method, we tune the value of λ from $\{1, 5, 10, 100, 500, 1000\}$.

Table 5: Implementation details.

Dataset	CIFAR-10	DomainNet	XGLUE-NC	QA
Model	CLIP	CLIP	XLM-Roberta-Base	LLaMA-2-7B
Batch size	64	64	32	16
Learning rate	0.01	0.01	0.01	2e-5
Local steps*	5	1	-1	-1
Total epochs	50	30	20	2
λ	1000	10	10	5

*The local steps -1 means clients iterate all training samples (i.e., one single local training epoch).