FedPFT: Federated Proxy Fine-Tuning of Foundation Models

Zhaopeng Peng[1] , **Xiaoliang Fan**[1,*] , **Yufan Chen**[1] , **Zheng Wang**[1] , **Shirui Pan**[2] , **Chenglu Wen**[1] , **Ruisheng Zhang**[3] , **Cheng Wang**[1]
[1]Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University
[2]School of Information and Communication Technology, Griffith University
[3]School of Information Science and Engineering, Lanzhou University
pengzhaopeng@stu.xmu.edu.cn, fanxiaoliang@xmu.edu.cn, {yufanchen, zwang}@stu.xmu.edu.cn, s.pan@griffith.edu.au, clwen@xmu.edu.cn, zhangrs@lzu.edu.cn, cwang@xmu.edu.cn

Abstract

Adapting Foundation Models (FMs) for downstream tasks through Federated Learning (FL) emerges a promising strategy for protecting data privacy and valuable FMs. Existing methods fine-tune FM by allocating sub-FM to clients in FL, however, leading to suboptimal performance due to insufficient tuning and inevitable error accumulations of gradients. In this paper, we propose Federated Proxy Fine-Tuning (FedPFT), a novel method enhancing FMs adaptation in downstream tasks through FL by two key modules. First, the sub-FM construction module employs a layer-wise compression approach, facilitating comprehensive FM fine-tuning across all layers by emphasizing those crucial neurons. Second, the sub-FM alignment module conducts a two-step distillations—layer-level and neuron-level—before and during FL fine-tuning respectively, to reduce error of gradient by accurately aligning sub-FM with FM under theoretical guarantees. Experimental results on seven commonly used datasets (i.e., four text and three vision) demonstrate the superiority of FedPFT. Our code is available at https://github.com/pzp-dzd/FedPFT.

1 Introduction

In recent years, various transformer-based Foundation Models (FMs) [Bommasani *et al.*, 2021] such as BERT [Kenton and Toutanova, 2019], GPT [Radford *et al.*,], LLaMA [Touvron *et al.*, 2023], and ViT [Dosovitskiy *et al.*, 2020] have attained state-of-the-art performance across a diverse range of natural language processing (NLP) and computer vision (CV) tasks, yet also face both data privacy and FM copyright concerns. For instance, a FM trained on medical data might inadvertently memorize sensitive patient information, and companies that own closed-source FMs may choose not to share FMs with the public. Federated Learning (FL) [McMahan *et al.*, 2017] offers a privacy-preserving approach for collaborative fine-tuning of FMs among multiple participants. This approach is increasingly promising for FM fine-tuning applications, ensuring the adaptation of downstream tasks without directly sharing client data and server FM.

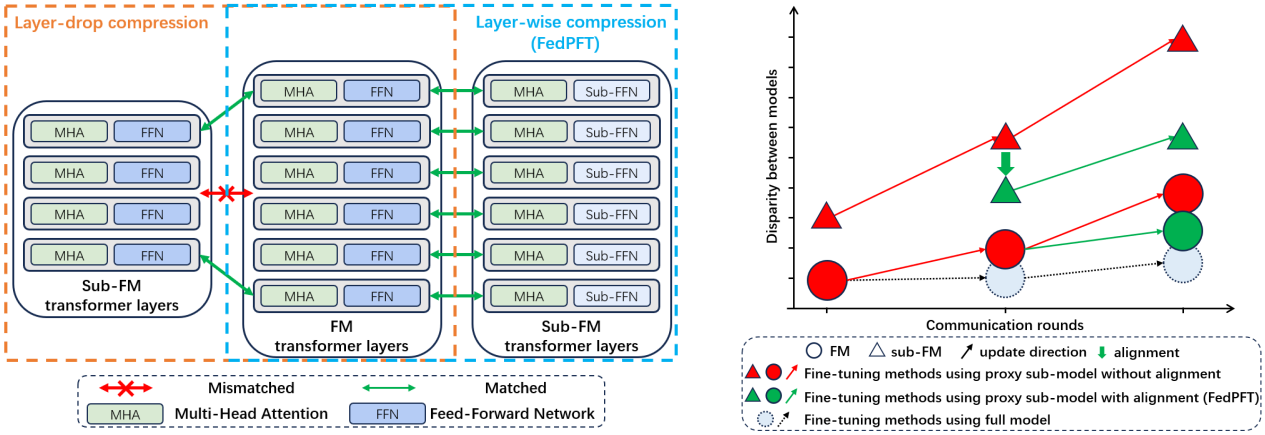
Recent methods [Xiao *et al.*, 2023; Marchisio *et al.*, 2023] mainly aim to fine-tune FMs without using the full model, which leverage layer-drop techniques [Sajjad *et al.*, 2023] to compress a FM and derive a sub-FM, enabling approximate fine-tuning of the original FM. However, these methods still pose **two significant challenges** that adversely reduce the performance of fine-tuned FMs. **On one hand**, they failed to fine-tune FMs sufficiently as a result of discarding those intermediate layers of FMs, consequently leading to the performance degradation of fine-tuned FMs. As shown in Fig.1(a), layer-drop methods fail to update intermediate layers of the FM during fine-tuning, due to the mismatch between the FM and the constructed sub-FM. **On the other hand**, there is a potential defect for the accumulation of gradient errors of FMs due to the lack of alignment between sub-FMs and FMs during FL fine-tuning, subsequently leading to further performance degradation. Fig.1(b) shows that, due to the absence of alignment, existing methods might accumulate significant gradients update errors between the FM and its constructed sub-FM during the FL fine-tuning process.

To address the above two challenges, we propose a framework called Federated Proxy Fine-Tuning (FedPFT) to enhance the adaptation of FMs for downstream tasks, while neither server FMs nor client data are directly shared. First, we design the sub-FM construction module, which performs layer compression on FMs to obtain sub-FMs by measuring neurons saliency of Feed-Forward Network (FFN) in transformer, facilitating comprehensive fine-tuning of FMs by emphasizing those crucial neurons. Second, we design the sub-FM alignment module, which conducts a two-step distillations—layer-level and neuron-level—before and during FL fine-tuning respectively, ensuring the accurate alignment between sub-FMs and FMs with a theoretical guarantee. Extensive experiments on three FMs and seven commonly used datasets demonstrate that FedPFT outperforms existing baselines that fine-tune FMs without using the full model.

Our contributions can be summarized as follows:

- We introduce FedPFT, a novel federated fine-tuning of FM method that establishes a sub-FM as a local proxy.

*Corresponding Author



(a) Two types of proxy sub-FM construction

(b) The problem of accumulating gradients errors

Figure 1: A motivating example of two challenges in FM fine-tuning using proxy sub-model. (a) Existing methods constructing sub-FMs via layer-drop compression discard intermediate layers in FM, causing mismatched and insufficient fine-tuning, while FedPFT conducting layer-wise compression ensures comprehensive fine-tuning of FM; and (b) as FL fine-tuning progresses, the discrepancy between the updates made by sub-FMs and FMs grows, leading to a deviation from the ideal update direction, while FedPFT aims to mitigate this gap by accurately aligning sub-FMs and FMs.

FedPFT effectively improves fine-tuning performance while maintaining the critical constraint that neither the server FM nor the client data is directly shared.

- We propose the first module for constructing sub-FMs through layer-wise compression. This technique maintains layer correspondence across sub-FMs and FMs, ensuring the comprehensive fine-tuning of FM layers while also considering the alleviation of training overhead.
- We propose the second module to align sub-FMs with FMs via a two-step distillation—layer-level and neuron-level—before and during FL fine-tuning respectively. Additionally, we offer theoretical insights into the significance of distillation for fine-tuning using sub-model.
- We conduct extensive experiments on three FMs and seven commonly used datasets. Results demonstrate that FedPFT consistently outperforms existing baselines.

2 Related Works

2.1 FM Fine-Tuning Through FL

Traditional centralized fine-tuning faces privacy concerns due to data sharing. Recent works [Chen *et al.*, 2023a; Yu *et al.*, 2023; Zhuang *et al.*, 2023] introduce the concepts of Federated Foundation Models, to alleviate privacy concerns. [Fan *et al.*, 2023; Kuang *et al.*, 2023] propose various Fed-LLM platforms to support federated training of LLMs. [Xu *et al.*, 2023] fine-tune FM via FL on mobile devices. [Chen *et al.*, 2023b] apply FM to federated multi-task learning. [Chen *et al.*, 2023c] save the communication cost during FL training through block-level parameters dropout. [Wang *et al.*, 2023a] reduce the communication and computation cost by training different layers of BERT in each round. [Zhang *et al.*, 2023] apply parameter-efficient fine-tuning (PEFT) methods to federated fine-tuning of FMs for privacy defense. However, most of aforementioned methods rely on sharing the server FM. This limitation may pose risks of FM copyright leakage and impose a substantial computational burden on clients.

2.2 FM Fine-Tuning Without Using the Full Model

Early PEFT methods, including Lora [Hu *et al.*, 2021], Adapter-Tuning [Houlsby *et al.*, 2019], and Prefix-Tuning [Li and Liang, 2021], focus on efficient fine-tuning of complete FMs by reducing the number of tunable parameters. Despite these efforts, the gradient computation for tunable parameters still needs backpropagation through the entire FM [Sung *et al.*, 2022]. Recently, Offsite-Tuning [Xiao *et al.*, 2023] is proposed to achieve fine-tuning without the full model. In this approach, the FM owner sends a light-weight adapter and an emulator constructed through layer-drop and knowledge distillation [Hinton *et al.*, 2015] to clients. Clients then fine-tune the adapter on their data with support from the emulator. The refined adapter is subsequently returned and incorporated into the full model for the fine-tuning process. Similarly, mini-model adaptation [Marchisio *et al.*, 2023] constructs a shallow mini-model from a fraction of the FM's parameters. However, those methods either discard significant amount of intermediate layers in FM or face the problem of gradient error accumulation, resulting in sub-optimal fine-tuning performance. Different from conventional methods, we construct sub-FMs based on layer-wise compression and mitigate gradient error accumulation by a two-step distillations.

3 FedPFT

3.1 Preliminary

Federated Learning

Given N parties $P_i (i = 1, ..., N)$, each party holds data D_i. Let $L(\cdot, \cdot)$ be the loss function. FL aims to train a machine learning model Θ using the dataset $D = \cup D_i (i = 1, ..., N)$

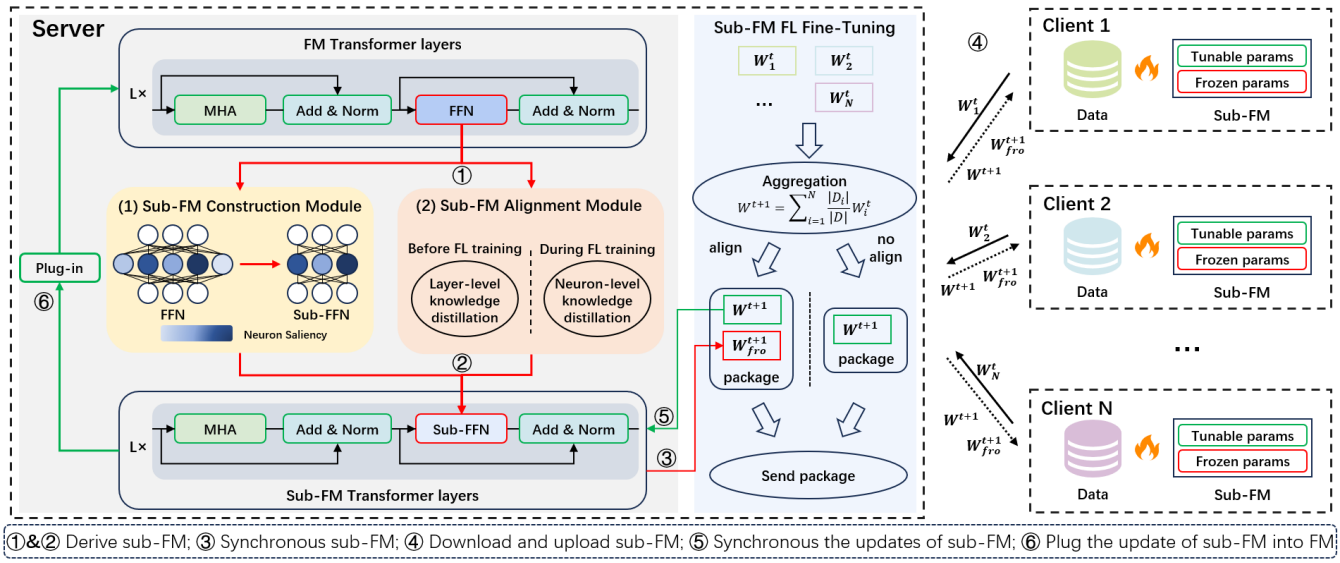


Figure 2: **The overall framework of FedPFT** that enhances FMs adaptation in downstream tasks through FL by two key modules: (1) Sub-FM Construction Module constructs sub-FM by layer-wise compression to facilitate comprehensive FM fine-tuning; and (2) Sub-FM Alignment Module aligns sub-FM by two-step distillation to ensure accurate alignment between sub-FM and FM with a theoretical guarantee.

under the coordination of a server S, while the raw data of all parties are not directly shared, which is formally described as

$$\Theta = \arg\min_{\Theta} \sum_{i=1}^{N} \frac{|D_i|}{|D|} L(\Theta, D_i). \tag{1}$$

Foundation Model Fine-tuning

Given a foundation model $\Theta = \{W_1, W_2, ..., W_n\}$ and a downstream task dataset D, the fine-tuning aims to obtain a new model $\Theta^* = \{W_1^*, W_2^*, ..., W_n^*\}$, it is

$$\begin{aligned} \Theta^* &= \Theta + \Delta\Theta, \\ \Delta\Theta &= \arg\min_{\Delta\Theta} L(\Theta + \Delta\Theta, D). \end{aligned} \tag{2}$$

3.2 Problem Definition

For FM fine-tuning using proxy sub-model, we first construct a sub-model $\Theta^{'} = \{W_1, W_2, ..., W_k, W_{k+1}^{'}, ..., W_n^{'}\}$ with fewer parameters for Θ to act as a proxy. Second, fine-tune the proxy sub-model $\Theta^{'}$ using the dataset D. Finally synchronize the updated gradients on $\Theta^{'}$ to Θ. Specifically, we construct $\Theta^{'}$ by compressing Θ, and retain a portion of the parameter matrix in Θ during the compression process. This compression process is formally described as follows:

$$\Theta^{'} = \Theta_1 \cup C(\Theta_2), \tag{3}$$

where $\Theta_1 \cup \Theta_2 = \Theta$, and $C(\cdot)$ denotes the compression method. During the fine-tuning of $\Theta^{'}$, we update only Θ_1 and synchronize the updated gradient on Θ_1 into Θ after fine-tuning to obtain Θ^* 's approximation of $\Theta^{*'}$, which is formally described as

$$\begin{aligned} \Theta^{*'} &= (\Theta_1 + \Delta\Theta_1^{'}) \cup \Theta_2, \\ \Delta\Theta_1^{'} &= \arg\min_{\Delta\Theta_1^{'}} L((\Theta_1 + \Delta\Theta_1^{'}) \cup C(\Theta_2), D). \end{aligned} \tag{4}$$

3.3 Method Overview

The overall framework of ours FedPFT is shown in Fig.2. We first derive a proxy sub-FM for the server FM, then collaboratively fine-tune the sub-FM through FL, and finally synchronise the updates on the sub-FM to the FM by plugging-in. FedPFT enhances downstream tasks adaptation of FMs through FL by two key module: (1) **Sub-FM Construction Module** that constructs sub-FMs by performing layer-wise compression on FMs based on neuron saliency; and (2) **Sub-FM Alignment Module** that reduces the difference between FMs and sub-FMs by layer-level and neuron-level knowledge distillation before and during FL fine-tuning, respectively. We will introduce those two modules in details as follows.

3.4 Sub-FM Construction Module Based on Layer-Wise Compression

Transformer-based FM typically consist of three parts: an embedding layer, a task head, and a sequence of transformer layers. Since the size of FM is dominated by all transformer layers, we perform compression for each transformer layer.

Each transformer layer contains two sub-layers: Multi-Head Attention (MHA) and Feed-Forward Network (FFN), each of which applies residual connection and followed by layer normalization. The output of MHA is

$$\begin{aligned} \mathrm{MHA}(x) &= \mathrm{Concat}(\mathrm{Attn}_0(x), ..., \mathrm{Attn}_h(x))W^O, \\ \mathrm{Attn}(x) &= \mathrm{softmax}(\frac{xW^Q(xW^K)^T}{\sqrt{d_k}})xW^V, \end{aligned} \tag{5}$$

where $W^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W^K \in \mathbb{R}^{d_{model} \times d_k}$, $W^V \in \mathbb{R}^{d_{model} \times d_k}$ and $W^O \in \mathbb{R}^{d_{model} \times d_{model}}$ are the weight matrices of query, key, value, and output in MHA, respectively. h is the number of attention heads, d_k and d_{model} are the dimensions of key and FM, respectively, and $d_{model} = d_k \times h$. The parameters number of MHA is about $4d_{model}^2$.

The output of FFN is

$$\mathrm{FFN}(x) = \mathrm{gelu}(xW_1 + b_1)W_2 + b_2, \tag{6}$$

where $W_1 \in \mathbb{R}^{d_{model} \times d_{ff}}$ and $W_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$ are the weight matrices of two linear layers in FFN, respectively, $b_1 \in \mathbb{R}^{d_{ff}}$ and $b_2 \in \mathbb{R}^{d_{model}}$ are the bias, d_{ff} is the dimensions of FFN and is usually set to $4 \times d_{model}$. The parameters number of FFA is about $8d_{model}^2$. Obviously, it is that most of the parameters in transformer layer are contained in FFN.

Hence, we opt to compress the FFN rather than the MHA of each layer for sub-FM construction. This minimizes the parameters number of sub-FM while ensuring a consistent set of trainable parameters (i.e. MHA) between the FM and its sub-FM at each layer. We accomplish layer-wise compression by systematically removing neurons with low saliency in the FFN of each layer, employing a fixed ratio.

First, by further transforming (6), we can represent the output of FFN as the sum of d_{ff} neurons outputs:

$$\mathrm{FFN}(x) = \sum_{i=1}^{d_{ff}} (\mathrm{gelu}(xu_i + b_{1i})w_i) + b_2, \tag{7}$$

where $w_i \in \mathbb{R}^{d_{model}}$ is the ith column vector in W_2, $u_i \in \mathbb{R}^{d_{model}}$ is the ith row vector in W_1, b_{1i} is the ith item in b_1.

Second, based on (7) and magnitude-based pruning method [Wen *et al.*, 2016], we use the L2-norm of all connect weights of neuron to measure its saliency, that is:

$$\mathrm{Saliency}(i) = \sqrt{\sum_{j=1}^{d_{model}} (w_{ij}^2 + u_{ij}^2)}, \tag{8}$$

where i is the index of neurons in FFN.

Finally, we construct a sub-FM serving as a proxy for the FM, accomplished by systematically eliminating neurons with low saliency in each layer at a fixed ratio.

3.5 Sub-FM Alignment Module Based on Two-Step Knowledge Distillation

In accordance with the description of FM fine-tuning using proxy sub-model in 3.2, it is evident that the FM fine-tuning is entirely contingent on the gradient descent of its sub-FM. This fine-tuning methodology prompts a fundamental question: How can we ensure the convergence of FM to the optimal value with the assistance of its sub-FM?

Theorem 1. *Suppose both the function $f : \mathbb{R}^n \to \mathbb{R}$ and its approximation $f' : \mathbb{R}^n \to \mathbb{R}$ are convex and differentiable, and their gradient are Lipschitz continuous with constant L_1>0 and L_2>0, respectively, i.e. we have that $\|\nabla f(x) - \nabla f(y)\|_2 \le L_1\|x-y\|_2$ and $\|\nabla f'(x) - \nabla f'(y)\|_2 \le L_2\|x-y\|_2$ for any x, y. Then if we run gradient descent for k iterations on f' with a fixed step size $\eta \le \frac{1}{L_1}$ and synchronize the gradient to f, let $\nabla f' - \nabla f = \delta$, when satisfying*

$$\|\delta\|_2^2 < \frac{1}{2}\|\nabla f\|_2^2, \tag{9}$$

$$\eta \sum_{i=1}^{k} \|\delta^{(i)}\|_2^2 \le \sum_{i=1}^{k} \langle \delta^{(i)}, x^{(i)} - x^* \rangle, \tag{10}$$

it will yield a solution $f^{(k)}$ which satisfies

$$f(x^{(k)}) - f(x^*) \le \frac{\|x^{(0)} - x^*\|_2^2}{2\eta k}, \tag{11}$$

where $f(x^*)$ is the optimal value.

Proof. See Appendix[1].A □

Intuitively, Theorem.1 indicates that when (9) and (10) are satisfied, gradient descent of FM with the help of sub-FM is guaranteed to converge and converges with rate $O(\frac{1}{k})$. It is evident that both conditions (9) and (10) are constraints on the difference between the actual and ideal update gradients of FM, and thus how to minimize the difference of the update gradients becomes a problem to be solved in the next step.

Theorem 2. *For a transformer, let the number of attention head be 1, and ignoring its nonlinear function and residual connection, its output can be expressed as $y = xW^Q(xW_K)^T xW^V W^O W_1 W_2$, let $A = W^Q(W^K)^T$, $B = W^V W^O$, $C = W_1 W_2$, then $y = xAx^T xBC$, and the output of its corresponding sub-layer after compressing FFN layer is expressed as $y' = xAx^T xBC'$, assuming that the gradient of loss function $loss = f(y)$ is Lipschitz continuous with constant L_3>0 and $\|C - C'\|_2^2 \le \epsilon_1$, $\|y - y'\|_2^2 \le \epsilon_2$, there exists the constant K_1>0 and K_2>0 such that*

$$\begin{aligned} \|\frac{\partial loss'}{\partial A} - \frac{\partial loss}{\partial A}\|_2^2 &\le K_1\epsilon_1 + K_2\epsilon_2, \\ \|\frac{\partial loss'}{\partial B} - \frac{\partial loss}{\partial B}\|_2^2 &\le K_1\epsilon_1 + K_2\epsilon_2, \end{aligned} \tag{12}$$

Proof. See Appendix[1].B □

From Theorem.2, it is evident that shrinking the error of gradients can be achieved by narrowing the difference in output and weights between the sub-FM and FM.

Based on the above analysis, we grasp the importance of narrowing the difference between sub-FM and FM via knowledge distillation to boost the performance of FM that fine-tuned using sub-FM. Therefore, we propose a method to align sub-FM using layer-level and neuron-level distillations in two phases, before and during FL fine-tuning, respectively. These two distillation methods are shown in the Fig.3.

Layer-Level Distillation Before FL Fine-Tuning

Given that our sub-FMs are constructed based layer-wise compression, where each layer retains a set of tunable parameters (i.e., MHA), we leverage the outputs from all layers to compute the layer-level distillation loss.

Furthermore, based on Theorem.2, we enhance the aforementioned distillation loss by introducing a regularization term. The purpose of this regularization term is to quantify the disparity between the weights of FFN and sub-FFN in each layer, to further facilitate a thorough knowledge transfer during fine-tuning by refining the alignment process. Thus,

[1]https://arxiv.org/abs/2404.11536

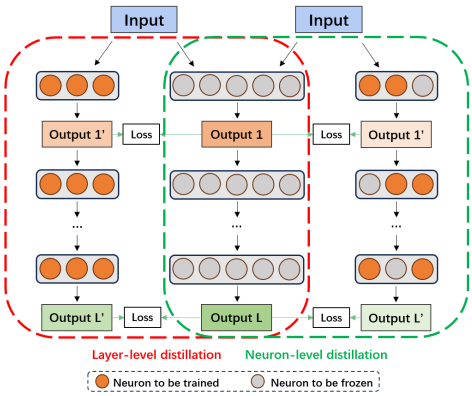


Figure 3: An example of two distillation processes

the final distillation loss is denoted as:

$$L_{KD} = \frac{1}{LM_{KD}} \sum_{i=1}^{L} ((\sum_{j=1}^{M_{KD}} \|O_j^{(i)} - O_j^{'(i)}\|_2^2) + \mu \|W_1^{(i)} W_2^{(i)} - W_1^{'(i)} W_2^{'(i)}\|_2^2), \quad (13)$$

where L is the number of layers, M_{KD} is the size of distill dataset, $O_j^{(i)}$ is the output of the jth sample in the ith layer, $W_1^{(i)}$ and $W_2^{(i)}$ are the two weight matrices of the ith FFN, μ is the regularization coefficient.

Neuron-Level Distillation During FL Fine-Tuning

In addition, the absence of alignment between FM and its constructed sub-FM during FL fine-tuning may cause the actual gradient update direction of the FM to gradually deviate from its ideal direction. This deviation can substantially reduce the performance of the fine-tuned FM. To mitigate the problem, we re-align the sub-FM with the FM after FL aggregation in certain rounds.

However, since the datasets for distillation and the datasets for local fine-tuning are typically collected from different domains, excessively aligning sub-FMs through distillation may hinder the adaptation of sub-FMs to downstream tasks. Inspired by [Mallya and Lazebnik, 2018], during the alignment process in FL fine-tuning, we opt to update only a subset of neurons with low saliency in local fine-tuning to prevent the risk of sub-FM forgetting knowledge of local data.

Moreover, since all FFNs of sub-FM are not updated during FL fine-tuning, the effectiveness of magnitude-based neuron saliency measurement methods diminishes. To address this, we opt to select neurons for updating during alignment based on the Average Percentage of Zero activations (APoZ) of outputs on the downstream task dataset [Hu *et al.*, 2016]. The $APoZ_k^{(i)}$ of the kth neuron in ith layer is defined as:

$$APoZ_k^{(i)} = \frac{1}{M_{DT} S} \sum_{j=1}^{M_{DT}} \sum_{l=1}^{S} \mathbb{I}(O_{jkl}^{(i)} = 0), \quad (14)$$

where M_{DT} is the size of the downstream task dataset, S is the sequence length of the jth sample, $O_{jkl}^{(i)}$ is the output of the lth token of the jth sample at the kth neuron in ith layer, $\mathbb{I}(\cdot)$ is the indicator function.

We calculate the APoZ for each neuron on the client using the local dataset before each round that requires alignment, and subsequently select the neurons that need to be updated during alignment based on their APoZ values.

3.6 Cost Analysis

We perform a theoretical analysis of the computational and communication cost of FedPFT based on BERT, and other models such as RoBERTa and ViT are similar. Following the settings in [Wang *et al.*, 2023a], we assume that all FL clients have the same training settings and exclude external differences such as local dataset size and hardware environment.

Computational Cost

Given a BERT model, let V be the vocabulary size, S be the sequence length, L be the number of layers, and c_f,c_b be the number of forward propagation and backward propagation respectively. Based on the analysis in 3.4, the computational cost of a BERT model is $O(d_{model}(V+S) + L(4Sd_{model}^2 + S^2 d_{model}) + 2LSd_{model}d_{ff} + LSd_{model})$ where the four terms denote the cost of embedding, MHA, FFN and Add&Norm, respectively.

Based on the above information, the overall time complexity of the full model is computed as follows. First, the time complexity of embedding is $O(d_{model}(V+S))$. Second, due to $d_{ff} = 4d_{model}$, the forward propagation time complexity is about $O(c_f L(12Sd_{model}^2 + S^2 d_{model}))$. Identically, the backward propagation time complexity is $O(c_b L(12Sd_{model}^2 + S^2 d_{model}))$. Finally, the overall time cost is $O(d_{model}(V+S) + L(c_f + c_b)(12Sd_{model}^2 + S^2 d_{model}))$ and we have $d_{model}(V+S) \ll L(c_f + c_b)(12Sd_{model}^2 + S^2 d_{model})$ and $S < d_{model}$.

In FedPFT, because $d'_{ff} = d_{model}$ for sub-FM, the time complexity during local training is $O(L(c_f + c_b)(6Sd_{model}^2 + S^2 d_{model}))$. Thus, compared with full model, FedPFT could reduce almost half the computational cost of all clients.

Communication Cost

Following [Vaswani *et al.*, 2017], the space complexity of a BERT model is $O(d_{model}(V+S) + 12Ld_{model}^2 + 2Ld_{model})$.

In FedPFT, if PEFT methods is not used, all model parameters need to be transmitted in each iteration, thus the communication cost will be in the complexity of $O(d_{model}(V+S) + 6Ld_{model}^2 + 2Ld_{model})$, again shrinking nearly half of the cost. If PEFT method is used, take Lora as an example, let t be the interval between two alignments during FL fine-tuning, q be the proportion of neurons that need to be updated during alignment, r be the rank of Lora, and Lora is only added to MHA, then the communication cost will be in the complexity of $O(8Lrd_{model} + \frac{2}{t} qLd_{model}^2)$. Since $\frac{1}{t} qd_{model}$ is usually smaller than r, the complexity is about $O(Lrd_{model})$, which does not impose a significant computational cost.

4 Experiments

4.1 Experimental Setting

Models and Datasets

Our evaluations span a variety of FMs, including two NLP FMs (i.e., BERT-base [Kenton and Toutanova, 2019],

Model	Setting	Method	SST-2	QNLI	MNLI-(m/mm)	QQP	Transformers Params
BERT	Fine-tuning with full model	FedPETuning	91.6	88.4	80.7/81.6	87.8	81M
	Fine-tuning without full model	FedOT	90.4	83.9	74.9/74.9	81.6	47M
		FedPFT	**91.6**	**87.5**	**78.6/79.0**	**86.3**	47M
RoBERTa	Fine-tuning with full model	FedPETuning	93.6	90.8	86.1/85.5	88.3	81M
	Fine-tuning without full model	FedOT	92.8	85.3	80.6/81.4	84.6	47M
		FedPFT	**93.1**	**88.7**	**83.4/83.2**	**87.1**	47M

Table 1: **Overall experimental results on BERT and RoBERTa.** The evaluation metric is accuracy. For MNLI, 'm' and 'mm' denote the matched and mismatched results, respectively. The best result for the same setting is marked in bold.

RoBERTa-base [Liu *et al.*, 2019]) and one CV FM (i.e., ViT-base [Dosovitskiy *et al.*, 2020]). Three FMs share consistent hyper-parameters for the number of layers L, attention heads h, hidden dimension d_{model}, and FFN dimension d_{ff}, all set at $L = 12$, $h = 12$, $d_{model} = 768$, and $d_{ff} = 3072$. Our NLP FM evaluations encompass four text datasets: SST-2 [Socher *et al.*, 2013], QNLI [Socher *et al.*, 2013], MNLI [Williams *et al.*, 2017], and QQP[2]. The CV FM is evaluated on three image datasets: CIFAR-10 [Krizhevsky *et al.*, 2009], CIFAR-100 [Krizhevsky *et al.*, 2009] and Flowers [Nilsback and Zisserman, 2008]. Specifically, SST-2 is the text sentiment analysis task, QNLI and MNLI are the natural language inference tasks, QQP is the text matching task, while CIFAR-10, CIFAR-100 and Flowers focus on image recognition tasks. In addition, we employ the Bookcorpus [Zhu *et al.*, 2015] and Wikipedia datasets for distillation of NLP sub-FMs, and the ImageNet-1k [Russakovsky *et al.*, 2015] for the distillation of CV sub-FM, respectively. Detailed dataset descriptions can be found in Appendix[1].C.

Baselines

We compare FedPFT with two methods, including: (1) FedPETuning [Zhang *et al.*, 2023] that performs parameter-efficient fine-tuning (PEFT) with full model through FL; and (2) FedOT that performs PEFT without full model through FL. It is worth noting that FedOT is the FL implementation of Offsite-Tuning [Xiao *et al.*, 2023] with multiple clients.

Hyper-parameters and Implementation

In the FL scenario, we set up 100 clients with 500 total communication rounds and employ the Dirichlet data partition method [Hsu *et al.*, 2019] to construct different label-skew data heterogeneity scenarios. In each communication round, we randomly select 10 clients for local fine-tuning, using a linear decay of the global learning rate over rounds and AdamW as the local fine-tuning optimizer. We use FedAvg [McMahan *et al.*, 2017] for global model aggregation. For FedOT, following [Xiao *et al.*, 2023], we use 2 layers at the bottom and 2 layers at the top as $Adapter$, and compress the intermediate 8 layers into 3 layers as $Emulator$. For FedPFT, we construct sub-FMs by performing layer-wise compression on the intermediate 10 layers. For fair comparison, we keep the number of trainable parameters the same for all three methods. The evaluation metric is the accuracy on the given validation set. All pre-trained models used in experiments are obtained from Hugging Face[3]. The FL scenarios were implemented using FLGo [Wang *et al.*, 2023b], and all experiments are conducted in PyTorch 2.1 and NVIDIA 3090 GPUs. Other detailed hyper-parameters can be seen in Appendix[1].C.

[2]https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

Model	Method	CIFAR-10	Transformers Params
ViT	FedPETuning	98.2	81M
	FedOT	95.5	47M
	FedPFT	**97.2**	47M

Table 2: **Experimental results on ViT**. The evaluation metric is accuracy.

4.2 Overall Comparisons

We first evaluate all three methods in Independent Identically Distributed (I.I.D) data distribution scenarios. Table.1 compares our FedPFT with two baselines for fine-tuning BERT and RoBERTa on four text datasets. We observe that: 1) the performance of all FMs fine-tuned by our FedPFT surpasses that achieved by FedOT and closely approaches the performance achieved by FedPETuning that fine-tuning using full model; and 2) FedOT exhibit a substantial performance gap compared to FedPETuning. This observed discrepancy from FedOT might be attributed to the absence of training the $Emulator$ and the accumulation of gradient errors during fine-tuning.

We further validate the effectiveness of FedPFT for fine-tuning CV FM. Table.2 presents the comparison results on the ViT model with CIFAR-10 dataset, and FedPFT still outperforms FedOT and achieves competitive performance closer to FedPETuning. The results on other two vision datasets (i.e., CIFAR-100, Flowers) are shown in Appendix[1].D.

4.3 Impact of Data Heterogeneity

We then evaluate the performance of FMs fine-tuned by three different methods in Non-Independent Identically Distributed (Non-I.I.D) data distribution scenarios. For the datasets used in the data heterogeneity experiments, we unequally partition

[3]https://huggingface.co/

Model	Method	SST-2			QNLI			Transformers Params
		Dir-1.0	Dir-5.0	Dir-10.0	Dir-1.0	Dir-5.0	Dir-10.0	
BERT	FedPETuning	90.6	90.9	91.1	87.3	87.9	88.1	81M
	FedOT	89.1	89.9	90.1	82.4	82.9	83.1	47M
	FedPFT	**89.7**	**91.1**	**91.6**	**86.1**	**86.9**	**87.2**	47M
RoBERTa	FedPETuning	93.0	93.3	93.5	90.2	90.6	90.8	81M
	FedOT	91.7	92.1	92.8	84.1	84.6	85.1	47M
	FedPFT	**92.1**	**92.7**	**93.1**	**87.2**	**87.6**	**88.1**	47M

Table 3: **Non-IID experimental results on BERT and RoBERTa**. The evaluation metric is accuracy.

Method	Model	
	BERT	RoBERTa
FedOT	83.9	85.3
FedPFT_N	83.0	78.9
FedPFT_B	86.9	86.7
FedPFT_D	84.3	85.5
FedPFT(ours)	**87.5**	**88.7**

Table 4: **Ablation study of FedPFT** on QNLI

Hyper-Parameter		Model	
		BERT	RoBERTa
Alignment interval t	5	87.3	88.4
	10	**87.5**	**88.7**
	20	87.1	87.1
Updated neurons proportion p	0.3	87.4	87.8
	0.5	**87.5**	**88.7**
	0.8	87.1	87.9

Table 5: **Parameter study of FedPFT** on QNLI

the dataset into 100 clients following the label distribution $Y_i \backsim Dir(\alpha p)$, where i is the client id, p is the global label distribution, α is the degree of Non-I.I.D and a smaller α generates a high label distribution shift. We construct three different label-skewing scenarios by adjusting the value of α: Dir-1.0, Dir-5.0, and Dir-10.0. Table.3 shows the comparison of three methods for fine-tuning BERT and RoBERTa in different data-heterogeneous scenarios. We observe that: 1) the performance of all methods declines as the degree of Non-I.I.D increases; 2) our FedPFT still outperforms FedOT and achieves competitive performance closer to FedPETuning. Visualisation of label distributions and results of Non-I.I.D experiments on vision datasets are shown in the Appendix[1].E.

4.4 Ablation Study

To evaluate the efficacy of individual components within FedPFT, we design the following variants for conducting ablation study:

- FedPFT_N which does not perform alignment by knowledge distillation;
- FedPFT_B which perform sub-FM alignment by knowledge distillation only before FL fine-tuning;
- FedPFT_D which perform sub-FM alignment by knowledge distillation only during FL fine-tuning;

Moreover, to validate the effectiveness of the sub-FM construction module of FedPFT, we list the experimental results of FedOT for comparison with FedPFT_B, as both perform sub-FM alignment by knowledge distillation only before FL fine-tuning. Results are shown in Table.4 and we observe that: 1) both FedPFT_N which does not align the sub-FMs entirely and FedPFT_D which lacks the alignment before FL fine-tuning, exhibit notably poor performance. This is consistent with our theoretical analysis (see Theorem.1), indicating that a significant disparity between the gradients of sub-FM and FM can impede the convergence of fine-tuning methods using proxy sub-model; 2) FedPFT_B outperforms FedOT, showing the effectiveness of the sub-FM construction module in FedPFT; and 3) FedPFT outperforms FedPFT_B, emphasizing the necessity of sub-FM alignment during FL fine-tuning. Results on other text and vision datasets are presented in Appendix[1].F.

4.5 Parameter Study

In addition, we conduct a study to investigate the impact of two hyper-parameters in the sub-FM alignment module: the interval t between two alignments during FL fine-tuning and the proportion p of neurons that need to be updated for each alignment. The effects of two hyper-parameters on the QNLI dataset are presented in Table.5, suggesting that both t and p should be chosen moderately. Regarding t, longer alignment intervals lead to the accumulation of gradient errors, while shorter intervals can prohibit the adaptation of downstream tasks. Concerning p, it is crucial to strike a balance between updating an adequate proportion of neurons and avoiding excessive disruption to local fine-tuning.

5 Conclusion

This paper introduces FedPFT, a federated fine-tuning framework designed for Foundation Models (FMs). FedPFT addresses critical challenges related to insufficient FM fine-tuning and the accumulation of gradient errors by employing layer-wise compression for sub-FM construction and aligning sub-FM through a two-step distillation process, respectively. This novel framework achieves optimal downstream task adaptation of FM, resulting in an effective fine-tuned FM with superior performance, all without direct sharing of either server FM or client data. Experimental results across seven datasets showcase the effectiveness of FedPFT. In the future, we aim to extend the application of FedPFT to larger-scale FMs for tackling more complex downstream tasks.

Acknowledgments

The research were supported by Natural Science Foundation of China (62272403) and the Fundamental Research Funds for the Central Universities (No. 20720220064).

References

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

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

[Chen *et al.*, 2023b] Yiqiang Chen, Teng Zhang, Xinlong Jiang, Qian Chen, Chenlong Gao, and Wuliang Huang. Fedbone: Towards large-scale federated multi-task learning. *arXiv preprint arXiv:2306.17465*, 2023.

[Chen *et al.*, 2023c] Yuanyuan Chen, Zichen Chen, Pengcheng Wu, and Han Yu. Fedobd: Opportunistic block dropout for efficiently training large-scale neural networks through federated learning. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 3541–3549. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[Fan *et al.*, 2023] Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. Fate-llm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049*, 2023.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Houlsby *et al.*, 2019] 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 *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

[Hsu *et al.*, 2019] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[Hu *et al.*, 2016] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

[Hu *et al.*, 2021] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

[Kenton and Toutanova, 2019] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Kuang *et al.*, 2023] 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.

[Li and Liang, 2021] 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 (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[Mallya and Lazebnik, 2018] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[Marchisio *et al.*, 2023] Kelly Marchisio, Patrick Lewis, Yihong Chen, and Mikel Artetxe. Mini-model adaptation: Efficiently extending pretrained models to new languages via aligned shallow training. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5474–5490, Toronto, Canada, July 2023. Association for Computational Linguistics.

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

[Nilsback and Zisserman, 2008] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference*

on computer vision, graphics & image processing, pages 722–729. IEEE, 2008.

[Radford *et al.*,] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[Sajjad *et al.*, 2023] Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429, 2023.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[Sung *et al.*, 2022] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *Advances in Neural Information Processing Systems*, 35:12991–13005, 2022.

[Touvron *et al.*, 2023] 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.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2023a] Xin'ao Wang, Huan Li, Ke Chen, and Lidan Shou. Fedbfpt: An efficient federated learning framework for bert further pre-training. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4344–4352. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.

[Wang *et al.*, 2023b] Zheng Wang, Xiaoliang Fan, Zhaopeng Peng, Xueheng Li, Ziqi Yang, Mingkuan Feng, Zhicheng Yang, Xiao Liu, and Cheng Wang. Flgo: A fully customizable federated learning platform. *arXiv preprint arXiv:2306.12079*, 2023.

[Wen *et al.*, 2016] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29, 2016.

[Williams *et al.*, 2017] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

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

[Xu *et al.*, 2023] Mengwei Xu, Yaozong Wu, Dongqi Cai, Xiang Li, and Shangguang Wang. Federated fine-tuning of billion-sized language models across mobile devices. *arXiv preprint arXiv:2308.13894*, 2023.

[Yu *et al.*, 2023] Sixing Yu, J Pablo Muñoz, and Ali Jannesari. Federated foundation models: Privacy-preserving and collaborative learning for large models. *arXiv preprint arXiv:2305.11414*, 2023.

[Zhang *et al.*, 2023] Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9963–9977, Toronto, Canada, July 2023. Association for Computational Linguistics.

[Zhu *et al.*, 2015] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

[Zhuang *et al.*, 2023] Weiming Zhuang, Chen Chen, and Lingjuan Lyu. When foundation model meets federated learning: Motivations, challenges, and future directions. *arXiv preprint arXiv:2306.15546*, 2023.