LLM at Network Edge: A Layer-wise Efficient Federated Fine-tuning Approach

Jinglong Shen[1], Nan Cheng[1]*, Wenchao Xu[2], Haozhao Wang[3], Yifan Guo[1], Jiajie Xu[1]
[1]School of Telecommunications Engineering, Xidian University
[2]Department of Computing, The Hong Kong Polytechnic University
[3]School of Computer Science and Technology, Huazhong University of Science and Technology
jlshen@stu.xidian.edu.cn, dr.nan.cheng@ieee.org, wenchao.xu@polyu.edu.hk,
hz_wang@hust.edu.cn, {guoyifan, xujiajie}@stu.xidian.edu.cn

Abstract

Fine-tuning large language models (LLMs) poses significant computational burdens, especially in federated learning (FL) settings. We introduce Layer-wise Efficient Federated Fine-tuning (LEFF), a novel method designed to enhance the efficiency of FL fine-tuning while preserving model performance and minimizing client-side computational overhead. LEFF strategically selects layers for fine-tuning based on client computational capacity, thereby mitigating the straggler effect prevalent in heterogeneous environments. Furthermore, LEFF incorporates an importance-driven layer sampling mechanism, prioritizing layers with greater influence on model performance. Theoretical analysis demonstrates that LEFF achieves a convergence rate of $\mathcal{O}(1/\sqrt{T})$. Extensive experiments on diverse datasets demonstrate that LEFF attains superior computational efficiency and model performance compared to existing federated fine-tuning methods, particularly under heterogeneous conditions.

1 Introduction

Large Language Models (LLMs)[2] have exhibited remarkable capabilities in various downstream tasks, including text generation Li et al. (2024), language translation Ranathunga et al. (2023), and question answering Yu et al. (2024). Their success is primarily attributed to their increasing model scale Bahri et al. (2024), with contemporary models scaling from billions of parameters (e.g., GPT-2) to hundreds of billions (e.g., GPT-4). While fine-tuning pre-trained LLMs on task-specific data is the de facto approach for adaptation, the privacy-sensitive nature of user-generated data poses challenges to centralized collection. Federated learning (FL) offers a solution by enabling distributed model training without requiring data centralization McMahan et al. (2017); Huang et al. (2024). In FL, models are fine-tuned directly on user devices, thereby preserving data privacy. However, the computational demands of fine-tuning such large models often surpass the capabilities of typical consumer devices. For instance, while typical consumer GPUs are often limited to 24GB of graphics memory, fine-tuning a GPT-3 model even with 16-bit precision requires approximately 326GB of memory.

Addressing the significant computational demands of LLMs, researchers have explored various mitigation strategies Han et al. (2024). parameter-efficient fine-tuning (PEFT) methods, including Adapter Houlsby et al. (2019), LoRA Hu et al. (2022), and prompt tuning Lester et al. (2021), have

*Corresponding Author.

[2]In this paper, we focus on transformer-based language models. For simplicity, we refer to transformer blocks as layers.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

emerged as promising approaches. These methods fine-tune a small number of additional or selected parameters while keeping the bulk of the pre-trained model frozen, thereby substantially reducing computational overhead. However, applying PEFT methods within FL settings encounters several critical challenges. First, the non-independent and identically distributed (non-IID) nature of client data is a primary concern Qi et al. (2023), often leading to substantial performance degradation Huang et al. (2023, 2022). This degradation can be particularly pronounced in federated PEFT compared to full fine-tuning paradigms Babakniya et al. (2023). Additionally, computational heterogeneity across participating devices impairs resource utilization efficiency due to the *straggler effect*, where overall training progress is constrained by the slowest clients Shen et al. (2024). While recent approaches, such as FedDSE, explore sub-model extraction based on neuron activation patterns for resource-constrained FL Wang et al. (2024a), maintaining model performance under non-IID data distributions remains a significant hurdle.

To address these challenges, we propose **L**ayer-wise **E**fficient **F**ederated **F**ine-tuning (LEFF), a method that aims to preserve the efficacy of full-parameter fine-tuning while significantly reducing computational overhead. LEFF enables clients to selectively fine-tune specific layers according to their computational capacity, while other layers remain frozen during local training. This flexibility allows resource-constrained clients to fine-tune a reduced set of layers, thereby mitigating the straggler effect. Following local training, clients transmit only the updated parameters of their selected layers to the server. The server then performs layer-wise aggregation to reconstruct the global model for the subsequent round. To further optimize the fine-tuning process, we introduce an importance-based layer sampling strategy. This strategy dynamically adjusts the selection probability of each layer based on its contribution to overall fine-tuning performance. This approach ensures that more impactful layers are prioritized for updates. Our key contributions are:

- **Novel Architecture:** We introduce LEFF, a federated fine-tuning framework designed to preserve full-parameter fine-tuning efficacy while substantially mitigating client-side computational overhead.
- **Robustness to Data Heterogeneity:** LEFF leverages full-parameter fine-tuning, enabling superior adaptation to non-IID data distributions across clients, outperforming conventional PEFT methods like LoRA.
- **Computational Heterogeneity Mitigation:** Our framework accommodates computational heterogeneity by allowing clients to dynamically adjust their local training workload according to their available resources, thereby mitigating the straggler problem.
- **Adaptive Layer Prioritization:** We develop an adaptive layer selection mechanism employing an importance-based sampling algorithm. This prioritizes updates for performance-critical layers, enhancing both training efficiency and model effectiveness.
- **Theoretical Guarantees:** We provide a rigorous convergence analysis, demonstrating that LEFF achieves a convergence rate of $\mathcal{O}(1/\sqrt{T})$.
- **Empirical Validation:** Extensive experiments demonstrate LEFF's capability to significantly reduce client-side computational burden while achieving superior fine-tuning performance under heterogeneous data and system conditions, outperforming existing state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature on FL and PEFT. Section 3 details our proposed LEFF framework and its key components. Section 4 provides a convergence analysis. Section 5 presents comprehensive experimental results and analyses. Finally, Section 6 summarizes our findings and discusses future work.

2 Related Work

PEFT methods are essential for adapting LLMs by alleviating the prohibitive computational costs of full fine-tuning. Techniques have evolved from updating parameter subsets Zaken et al. (2021) and injecting trainable modules Houlsby et al. (2019) to latency-free approaches like LoRA Hu et al. (2022); Fu et al. (2022), which merges learned low-rank matrices into the original model. Further refinements, such as dynamically updating important layers as demonstrated by LISA Pan et al. (2024), optimize efficiency by targeting model capacity more effectively. While established in centralized

settings, PEFT's application to FL is an emerging research area Zhuang et al. (2023); Yu et al. (2023); Woisetschläger et al. (2024). Various PEFT strategies, including LoRA-based instruction tuning Zhang et al. (2024), federated prompt optimization Guo et al. (2023, 2024), and backpropagation-free client fine-tuning Xu et al. (2023), have shown viability in FL Chen et al. (2022, 2023); Sun et al. (2022); Zhang et al. (2023); Fang et al. (2024); Legate et al. (2023); Wang et al. (2024b). However, a critical challenge is data heterogeneity across clients, which analyses confirm significantly degrades PEFT performance compared to full fine-tuning Babakniya et al. (2023); Bai et al. (2024); Cho et al. (2023), a limitation even for specialized approaches like FedDAT in multimodal tasks Chen et al. (2024). Concurrently, federated full-parameter tuning methods present their own trade-offs, including high client computation in gradient approximation Qin et al. (2024), unreduced peak memory load in update-compression schemes Shu et al. (2025), and a lack of adaptability in static cyclical updates Wang et al. (2024c). To address this performance degradation, we propose LEFF. Unlike prior work, LEFF facilitates decentralized, layer-wise adaptation of backbone models by enabling client-specific full-parameter updates to a selected subset of layers, aiming to effectively mitigate the adverse impacts of data heterogeneity while preserving computational efficiency.

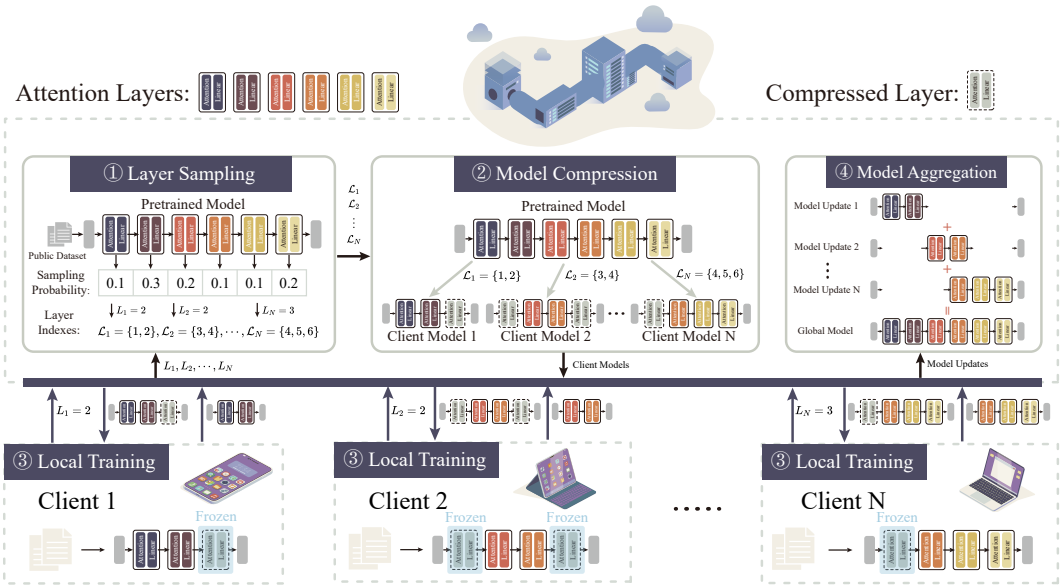

Figure 1: Overview of the LEFF framework. Each communication round comprises four key stages.

3 Methodology

3.1 System Architecture

FL clients in edge computing (e.g., mobile phones, personal computers) often possess limited computational resources. The increasing scale of LLMs intensifies this constraint, making their fine-tuning computationally burdensome. To mitigate this, we introduce LEFF, a framework to reduce client-side training overhead while striving to maintain the benefits of full-parameter fine-tuning.

The LEFF framework (Figure 1) involves four steps per communication round. First, clients report their local fine-tuning capacity (number of layers L_i). Based on these reports, the server calculates sampling probabilities and selects a layer subset $\mathcal{L}_i$ (of size L_i) for each client $\mathcal{C}_i$ to fine-tune. Second, to reduce client burden, the server compresses unselected layers $\mathcal{L}_i^-$ via knowledge distillation, creating a customized local model. This model comprises client-specific fine-tunable layers (from $\mathcal{L}_i$) and frozen compressed layers (derived from $\mathcal{L}_i^-$). Third, clients fine-tune only their assigned layers; other compressed layers remain frozen. Finally, clients upload their fine-tuned layer parameters, and the server performs layer-wise aggregation to update the global model.

We consider a standard FL setup: a central server aggregates parameters from N clients ($\mathcal{C}_i, i = 1, \ldots, N$) performing local training on their datasets $\mathcal{D}_i$ (size $D_i = |\mathcal{D}i|$). The global model Θ_g has L layers. For each client $\mathcal{C}i$, the server samples a block of L_i consecutive layers for fine-tuning,

denoted $\mathcal{L}_i$:

$$\mathcal{L}_i = \{l_{\text{start}}, \ldots, l_{\text{end}}\}, \quad \text{s.t.} \quad l_{\text{end}} - l_{\text{start}} + 1 = L_i. \tag{1}$$

The unselected layers $\mathcal{L}_i^- = \{l_k \mid 1 \leq l_k \leq L, l_k \notin \mathcal{L}_i\}$ are compressed by the server into $\check{\Theta}_g^{\mathcal{L}_i^-}$. The resulting client-specific model $\Theta_i = \{\Theta_g^{\mathcal{L}_i}, \check{\Theta}_g^{\mathcal{L}_i^-}\}$ supports forward propagation. Client $\mathcal{C}_i$ receives Θ_i, fine-tunes only $\Theta_g^{\mathcal{L}_i}$ on $\mathcal{D}_i$ to obtain updated parameters $\Theta_i^{\mathcal{L}_i}$ (while $\check{\Theta}_g^{\mathcal{L}_i^-}$ remain frozen), and uploads $\Theta_i^{\mathcal{L}_i}$. The server aggregates these layer-specific parameters from all clients to update Θ_g.

3.2 Layer Sampling

At the start of each communication round, the server selects the layers for each client to fine-tune based on L_i. Considering that layers contribute differently to fine-tuning performance, we propose an importance-based layer sampling method. This approach prioritizes layers considered more crucial for model fine-tuning, thereby optimizing the fine-tuning process by concentrating computational resources on the most impactful model parts.

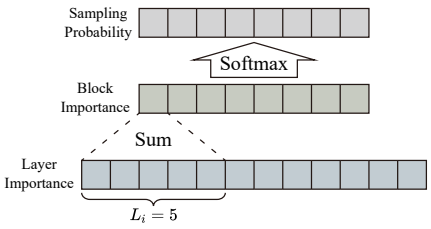

Figure 2: Sampling probability for each block. Model layers are organized into blocks, and clients sample only one block per communication round for fine-tuning.

As discussed in Molchanov et al. (2019), the importance of a neural parameter can be quantified by the change in the loss value when the parameter is introduced or removed:

$$\mathcal{I}_m = \left(\mathcal{F}(\mathcal{D}, \Theta) - \mathcal{F}(\mathcal{D}, \Theta \mid_{\theta_m=0})\right)^2, \tag{2}$$

where $\mathcal{F}$ denotes the loss function, $\mathcal{D}$ represents the dataset, and θ_m is the m-th parameter of the model Θ. We further adopt its first-order Taylor expansion form:

$$\mathcal{I}_m^{(1)}(\Theta) = (g_m \theta_m)^2, \tag{3}$$

where g_m is the gradient of the m-th parameter. This approximation allows for estimating each parameter's importance with only a single forward and backward pass, significantly reducing computational complexity.

As illustrated in Figure 2. Based on the importance scores of individual parameters, the importance score for each layer can be derived by summing the importance scores of all parameters within that layer:

$$\mathcal{I}_{\Theta^l} \approx \mathcal{I}_{\Theta^l}^{(1)}(\Theta) = \sum_{m \in \Theta^l} \mathcal{I}_m^{(1)}(\Theta) = \sum_{m \in \Theta^l} (g_m \theta_m)^2 \tag{4}$$

We define consecutive L_i layers as a *block*, denoted as $\bar{\Theta}^k = \{\Theta^k, \cdots, \Theta^{k+L_i-1}\}$, with its importance given by:

$$\mathcal{I}_{\bar{\Theta}^k} = \sum_{l=k}^{k+L_i-1} \mathcal{I}_{\Theta^l} \tag{5}$$

To determine the sampling probability, these importance scores are normalized using the Softmax function. Thus, the sampling probability for each block during layer sampling for client $\mathcal{C}_i$ is expressed as:

$$\mathbf{p}_i = \text{Softmax}\left(\{\mathcal{I}_{\bar{\Theta}^k} \mid k = 1, \cdots, L - L_i + 1\}\right) \tag{6}$$

The server utilizes $\mathbf{p}_i$ to sample a block for client $\mathcal{C}_i$, determining the set of layers $\mathcal{L}_i$ that the client will fine-tune in the current communication round.

3.3 Model Compression

As illustrated in Figure 3, in each communication round, prior to client-side fine-tuning, the server compresses the layers not selected for client fine-tuning, denoted by $\mathcal{L}_i^-$, to reduce computational overhead. Specifically, the server employs a proxy dataset $\mathcal{D}_{\text{proxy}}$, sourced from publicly available data, for this knowledge distillation process. The student model $\check{\Theta}_g^{\mathcal{L}_i^-}$ maintains a multi-layer attention architecture identical to that of the corresponding layers in the pre-trained global model. The number of layers in the student model is determined by a compression rate r. If the teacher model segment $\Theta_g^{\mathcal{L}_i^-}$ comprises $\left|\Theta_g^{\mathcal{L}_i^-}\right|$ layers, the student model's layer count is $\left|\check{\Theta}_g^{\mathcal{L}_i^-}\right| = \left\lceil \left|\Theta_g^{\mathcal{L}_i^-}\right| \cdot r \right\rceil$.

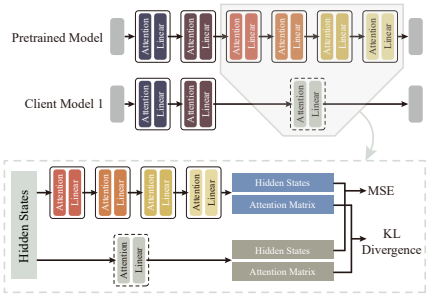

Figure 3: Distillation process in LEFF. The distillation loss comprises two components: the mean square loss of the hidden state and the KL divergence of the attention matrix.

Once the student model's architecture is defined, the server initiates compression via knowledge distillation. This distillation leverages two objectives: a Mean Squared Error (MSE) loss between the hidden states of the teacher and student models, and a Kullback-Leibler (KL) divergence between their attention matrices. Crucially, to prevent the student model from learning task-irrelevant information from $\mathcal{D}_{\text{proxy}}$, only the intermediate representations (hidden states and attention matrices) are used for distillation, not the task-specific output labels from $\mathcal{D}_{\text{proxy}}$. Specifically, for an input hidden state sequence H_i, the teacher model segment $\Theta_g^{\mathcal{L}_i^-}$ produces outputs $(H_t, A_t) = \Theta_g^{\mathcal{L}_i^-}(H_i)$, while the student model $\check{\Theta}_g^{\mathcal{L}_i^-}$ produces $(H_s, A_s) = \check{\Theta}_g^{\mathcal{L}_i^-}(H_i)$. The distillation loss E_{distill} is then formulated as:

$$E_{\text{distill}} = (1-\alpha)E_{\text{hidden\_state}} + \alpha E_{\text{attention\_matrix}} = (1-\alpha)\text{MSE}(H_t, H_s) + \alpha \text{KL}(A_t, A_s), \tag{7}$$

where α is a hyperparameter balancing the two loss components. Following compression, the server integrates the student model $\check{\Theta}_g^{\mathcal{L}_i^-}$ with the layers selected for fine-tuning, $\Theta_g^{\mathcal{L}_i}$, to form the client-specific model $\Theta_i = \{\Theta_g^{\mathcal{L}_i}, \check{\Theta}_g^{\mathcal{L}_i^-}\}$. This model Θ_i is then dispatched to the i-th client.

3.4 Local Training

Upon receiving the model Θ_i from the server, client $\mathcal{C}_i$ fine-tunes the selected layers $\Theta_g^{\mathcal{L}_i}$, while the unselected layers $\check{\Theta}_g^{\mathcal{L}_i^-}$ remain frozen to maintain the correct training context. This strategy enables clients to update only a subset of parameters, with the compressed unselected parameters requiring minimal resources. As a result, LEFF significantly reduces computational overhead on the client side, and adapts to the client's computational capabilities, thereby enhancing efficiency. After completing local training, client $\mathcal{C}_i$ sends the updated model parameters $\Theta_i^{\mathcal{L}_i}$ back to the server for aggregation.

3.5 Model Aggregation

After receiving model parameters from all clients, the server aggregates them to construct a comprehensive global model. Since each client fine-tunes different components of the model, we utilize a layer-wise aggregation approach, performing weighted averaging for each layer individually. Specifically, for a global model $\Theta_g = \{\Theta_g^1, \cdots, \Theta_g^l, \cdots, \Theta_g^L\}$ with L layers, let $\mathcal{S}_l$ represent the set of clients that fine-tuned the l-th layer. The aggregated global model is expressed as follows:

$$\Theta_g = \left\{\Theta_g^l \mid l = 1, \cdots, L\right\} = \left\{\sum_{i \in \mathcal{S}_l} \frac{D_i}{\sum_{j \in \mathcal{S}_l} D_j} \Theta_i^l \mid l = 1, \cdots, L\right\}. \tag{8}$$

Once the server completes the aggregation and acquires the updated global model, the subsequent communication round commences. This process continues until the model converges or reaches a predetermined number of iterations.

4 Convergence Analysis

4.1 Assumptions

Our convergence analysis of LEFF relies on the following assumptions. Assumptions 1,2,3 are standard in FL (Li et al., 2020); Assumption 4 addresses complexities from LEFF's layer selection and model compression.

Assumption 1 (L-Smoothness)**.** *Each client's local loss function $\mathcal{F}_i : \mathbb{R}^d \to \mathbb{R}$ is L-smooth with respect to the full model parameters Θ. Thus, for a constant $L > 0$, all $\Theta_1, \Theta_2 \in \mathbb{R}^d$, and all clients $i \in [N]$:*

$$\|\nabla \mathcal{F}_i(\Theta_1) - \nabla \mathcal{F}_i(\Theta_2)\| \leq L\|\Theta_1 - \Theta_2\|. \tag{9}$$

The global loss function $\mathcal{F}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{F}_i(\Theta)$ is thus also L-smooth.

Assumption 2 (Bounded Variance)**.** *The variance of stochastic gradients computed by each client is bounded. Let $\nabla \mathcal{F}_i(\Theta; \xi)$ be client i's stochastic gradient for sample $\xi \sim \mathcal{D}_i$ at parameters Θ, and $\nabla \mathcal{F}_i(\Theta) = \mathbb{E}_{\xi \sim \mathcal{D}_i}[\nabla \mathcal{F}_i(\Theta; \xi)]$ be the true local gradient. We assume constants $\sigma_i^2 \geq 0$ such that for all $\Theta \in \mathbb{R}^d$ and $i \in [N]$:*

$$\mathbb{E}_{\xi \sim \mathcal{D}_i}\|\nabla \mathcal{F}_i(\Theta; \xi) - \nabla \mathcal{F}_i(\Theta)\|^2 \leq \sigma_i^2. \tag{10}$$

We define $\sigma^2 = \max_{i \in [N]} \sigma_i^2$. This is standard in analyzing stochastic optimization, including FL.

Assumption 3 (Bounded Heterogeneity)**.** *To account for data heterogeneity (non-IID data), we assume the dissimilarity between true local and global gradients is bounded. For a constant $\zeta^2 \geq 0$ and all $\Theta \in \mathbb{R}^d$:*

$$\frac{1}{N}\sum_{i=1}^{N} \|\nabla \mathcal{F}_i(\Theta) - \nabla \mathcal{F}(\Theta)\|^2 \leq \zeta^2. \tag{11}$$

This is standard in FL analysis, especially for non-IID settings.

During local training in round t, client i performs K steps of stochastic gradient descent, starting from $\Theta_{i,t,0}^{\mathcal{L}_i} = \Theta_{g,t}^{\mathcal{L}_i}$. At local step $k \in \{0, \ldots, K-1\}$, the client updates $\Theta_{i,t,k}^{\mathcal{L}_i}$ using a stochastic gradient $G_{i,t,k}$ computed with respect to these active parameters:

$$G_{i,t,k} = \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{i,t,k}^{\mathcal{L}_i} | \check{\Theta}_{g,t}^{\mathcal{L}_i^-}; \xi_{i,t,k}). \tag{12}$$

Here, $\mathcal{F}_i(\cdot|\cdot; \xi)$ indicates the loss evaluated with active parameters $\Theta_{i,t,k}^{\mathcal{L}_i}$ conditioned on the fixed context $\check{\Theta}_{g,t}^{\mathcal{L}_i^-}$, using data sample $\xi_{i,t,k} \sim \mathcal{D}_i$.

Let $\bar{G}_{i,t}$ be the average expected effective gradient computed by client i over its K local steps in round t:

$$\bar{G}_{i,t} = \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}_{\xi_{i,t,k} \sim \mathcal{D}_i}[G_{i,t,k}]. \tag{13}$$

Let $\nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t})$ denote the restriction of the full local gradient $\nabla \mathcal{F}_i(\Theta_{g,t})$ to the subspace corresponding to the selected layers $\mathcal{L}_i$.

Assumption 4 (Bounded Model Approximation Error)**.** *The expected squared norm difference between the average effective gradient $\bar{G}_{i,t}$ and the corresponding part of the true local gradient $\nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t})$ (evaluated at the start-of-round global model Θ_t^g) is bounded. The expectation $\mathbb{E}_{\mathcal{L}_i \sim \boldsymbol{p}_i}$ is taken over the randomness of layer selection for client i (with selection distribution $\boldsymbol{p}_i$):*

$$\mathbb{E}_{\mathcal{L}_i \sim \boldsymbol{p}_i}\left[\|\bar{G}_{i,t} - \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t})\|^2\right] \leq \Delta_{i,t}^2. \tag{14}$$

Furthermore, we assume the average of these client-specific error bounds across all clients is bounded, potentially depending on the round t:

$$\frac{1}{N}\sum_{i=1}^{N} \Delta_{i,t}^2 \leq \Delta_t^2. \tag{15}$$

For simplicity in certain analyses, one might further assume a uniform bound Δ^2 such that $\Delta_t^2 \leq \Delta^2$ for all t.

4.2 Main Theorem

We present the key lemma leading to our main convergence theorem.

Lemma 1 (Bound on Expected Squared Norm of Global Update)**.** *Let Assumptions 1, 2, 3, and 4 hold. Let $\mathcal{A}_t$ be the set of clients selected in round t. The expected squared norm of the global model change in one communication round t is bounded by:*

$$\mathbb{E}\left[\left\|\Theta_{t+1}^g - \Theta_t^g\right\|^2\right] \leq C_3\eta^2K^2\left(\sigma^2 + \Delta_t^2 + \zeta^2 + \|\nabla\mathcal{F}(\Theta_t^g)\|^2\right), \tag{16}$$

where the expectation $\mathbb{E}[\cdot]$ is taken over the randomness of client selection $\mathcal{A}_t$, layer selections $\{\mathcal{L}_i\}_{i\in\mathcal{A}_t}$, and local stochastic gradients $\{\xi_{i,t,k}\}_{i\in\mathcal{A}_t,0\leq k<K}$. Here, C_3 is a constant that may depend on the total number of layers L, client sampling strategy, layer sampling probabilities $\boldsymbol{p}_i$, and aggregation weights.

Theorem 1 (Convergence of LEFF)**.** *Let Assumptions 1, 2, 3, and 4 hold. If η is chosen such that $\eta \leq C_4/(LC_3K)$ for sufficiently small C_4 (where C_3 is from Lemma 1, L is the smoothness constant from Assumption 1, and K is the number of local steps), then the LEFF algorithm satisfies:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{F}(\Theta_{g,t})\|^2] \leq \underbrace{\frac{C_8(\mathcal{F}(\Theta_{g,0}) - F^*)}{\eta KT}}_{\textit{Vanishing term}} + \underbrace{C_9\left(\bar{\Delta}^2 + \zeta^2 + \frac{\sigma^2}{K}\right) + C_{10}\eta K(\sigma^2 + \bar{\Delta}^2 + \zeta^2)}_{\textit{Error floor}}, \tag{17}$$

where $\mathbb{E}[\cdot]$ takes expectation over all randomness up to round t, T is the total number of communication rounds, $\bar{\Delta}^2 = \frac{1}{T}\sum_{t=0}^{T-1}\Delta_t^2$ is the average model approximation error over T rounds, F^* is a lower bound for the global loss function $\mathcal{F}(\Theta)$, and C_8, C_9, C_{10} are positive constants depending on problem parameters (L, σ^2, ζ^2), algorithm parameters (K), and constants from the assumptions and lemmas.

With an appropriately chosen decaying learning rate $\eta = \mathcal{O}(1/\sqrt{T})$ (satisfying the condition in Theorem 1 for sufficiently large T), the bound in Eq. (17) implies a convergence rate where the average squared gradient norm diminishes towards an error floor. This leads to the following corollary regarding the minimum expected gradient norm:

Corollary 1 (Convergence Rate)**.** *Under the conditions of Theorem 1, if a decaying learning rate $\eta = \mathcal{O}(1/\sqrt{T})$ is used, then LEFF achieves:*

$$\min_{0\leq t\leq T-1}\mathbb{E}[\|\nabla\mathcal{F}(\Theta_{g,t})\|^2] \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + C_9\left(\bar{\Delta}^2 + \zeta^2 + \frac{\sigma^2}{K}\right). \tag{18}$$

The convergence is guaranteed only up to an error floor, primarily determined by the non-vanishing term $C_9(\bar{\Delta}^2 + \zeta^2 + \sigma^2/K)$ in Eq. (18). This term quantifies the sources of residual error inherent in the federated optimization process with LEFF.

Trade-offs: The analysis reveals a fundamental trade-off inherent in LEFF. The framework enhances client-side efficiency (computation, memory, potentially comddddmunication) through layer selection and context compression. However, this introduces the model approximation error $\bar{\Delta}^2$, which contributes directly to the convergence error floor and may limit the achievable model accuracy. Consequently, the practical success of LEFF hinges on implementing layer selection and compression techniques that effectively minimize $\bar{\Delta}^2$ while preserving the desired efficiency gains.

5 Evaluations

5.1 Experimental Setup

We conduct extensive experiments to evaluate our proposed method against established FL approaches. The evaluation framework incorporates baseline algorithms, pre-trained models, benchmark datasets, and detailed experimental configurations.

We benchmark our approach against FedAvg McMahan et al. (2017), which averages full model parameters, and three parameter-efficient FL methods: FedBitFit Zaken et al. (2021), updating only

bias terms; FedLoRA Hu et al. (2022), employing low-rank updates; and SLoRA Babakniya et al. (2023), utilizing server-side singular value decomposition (SVD) for enhanced stability and efficiency. Experiments are conducted using GPT-2 Medium Radford et al. (2019) for natural language generation (NLG) and DeBERTaV3 Base He et al. (2021) for natural language understanding (NLU). We evaluate on the GLUE benchmark Wang et al. (2019) (e.g., CoLA, MRPC, MNLI) for NLU tasks and the E2E NLG Challenge Novikova et al. (2017) for NLG from structured meaning representations.

Experiments were conducted for 20 communication rounds on the public datasets WebNLG Gardent et al. (2017) and WNLI Wang et al. (2019). To simulate heterogeneous (non-IID) data distributions, client data was partitioned using a Dirichlet distribution with a concentration parameter (α) ranging from 0.05 to 50.0. Our FL simulations involved a varying number of clients, each performing one local training epoch per communication round. During local training, clients fine-tuned their local models using the AdamW optimizer Loshchilov, Hutter (2019) with a learning rate of 1×10^{-5}. All experiments were performed on a system with eight NVIDIA H100 GPUs. This experimental setup enables a systematic evaluation of the proposed method's robustness against diverse data distributions and varying client participation levels.

5.2 Results

5.2.1 Comparison with Baselines

We evaluate LEFF on the GLUE benchmark across varying data heterogeneity, controlled by the Dirichlet parameter $\alpha \in \{0.05, 0.1, 0.5, 1.0, 5.0, 10.0, 50.0\}$ (Figure 4). While FedAvg's full-parameter fine-tuning ensures robust performance across all α values, methods relying on partial parameter updates (e.g., FedBitFit, FedLoRA, and SLoRA) exhibit limitations, particularly in highly heterogeneous settings (low α). In contrast, LEFF employs a layer-wise federated fine-tuning strategy, dynamically selecting layers based on client computational capabilities and importance sampling. This approach enables LEFF to achieve consistently strong performance, effectively addressing both data and system heterogeneity and outperforming baselines in critical heterogeneity ranges. Notably, at high data heterogeneity ($\alpha = 0.05$, Table 1), FedAvg attains the highest GLUE scores, with LEFF delivering closely comparable results and also demonstrating competitive performance on NLG tasks.

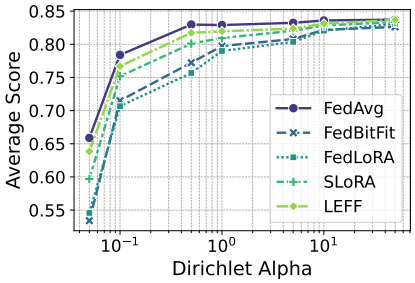

Figure 4: Average score of DeBERTaV3 on GLUE benchmark.

Table 1: Test results of DeBERTaV3 (trained on GLUE tasks) and GPT2 (trained on E2E NLG task). The best result per task group is marked in underline, and the secondary is marked in **bold**.

	DeBERTaV3 (GLUE Tasks)									GPT2 (E2E NLG Task)					
Model	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	Avg.	BLEU	NIST	METEOR	ROUGE	CIDEr	Avg.
FedAvg	28.36	81.08	71.45	87.07	73.24	68.43	65.26	52.17	65.88	0.5878	8.0645	0.4123	0.6405	1.7848	2.2980
FedBitFit	4.65	70.30	69.49	71.64	62.02	43.45	59.82	45.90	53.41	0.5570	7.3859	0.3443	0.5917	1.4059	2.0570
FedLoRA	6.31	70.49	68.38	78.40	68.97	41.97	58.97	43.07	54.57	0.5402	7.0344	0.3625	0.5857	1.3712	1.9788
SLoRA	21.82	72.44	69.00	80.35	69.08	53.26	60.68	50.97	59.70	0.5700	7.9592	0.4001	0.6243	1.7322	2.2572
LEFF	**27.74**	**79.30**	**70.00**	**85.00**	**70.83**	**62.75**	**63.24**	**51.89**	**63.84**	**0.5799**	**8.0296**	**0.4064**	**0.6346**	**1.7521**	**2.2805**

5.2.2 Effect of Compression Ratio

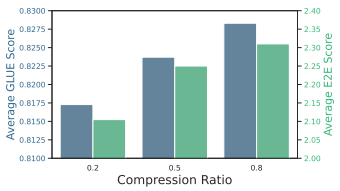

Figure 5: Average scores of LEFF on GLUE benchmark and E2E NLG dataset at different compression rates.

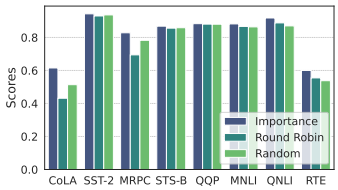

Figure 6: Scores of LEFF on GLUE benchmark using different sampling method.

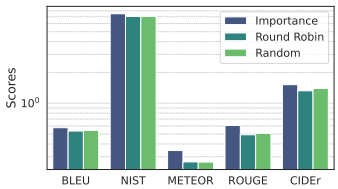

Figure 7: Scores of LEFF on E2E NLG dataset using different sampling method.

We analyze the sensitivity of LEFF's performance to the compression ratio r. Figure 5 shows that LEFF achieves superior performance with higher values of r (i.e., lower compression levels where more parameters are retained). With higher r, the student model retains more parameters, enhancing its representational capacity to emulate the teacher model more effectively. This, in turn, provides a more precise training context for the local model. Conversely, lower values of r (i.e., higher compression levels) indicate the student model retains fewer parameters, thereby constraining its representational capacity. This leads to less accurate teacher emulation and subsequently degrades LEFF's training efficacy and overall performance.

5.2.3 Effect of Client Scale

We evaluated model performance across a range of federated clients (8 to 40) on tasks from the GLUE benchmark and for NLG. Our results reveal a systematic degradation in performance as the number of clients increases. As detailed in Table 2, for GLUE tasks, the robustness to client scaling varied significantly across different metrics. For instance, SST-2 demonstrated minimal degradation, with performance decreasing by 3.10% (from 94.30 to 91.38). In contrast, CoLA exhibited substantial sensitivity, with its score declining by 49.18% (from 61.51 to 31.26). Consequently, the average GLUE score dropped from 81.73 to 72.37. NLG tasks, presented in Table 3, showed greater resilience. BLEU scores experienced a marginal reduction of 1.94% (from 0.5762 to 0.5650), while CIDEr registered the largest relative decline at 5.16% (from 1.5174 to 1.4391). These findings suggest that while the model can maintain reasonable efficacy in federated settings with fewer clients, its performance is challenged when scaling to larger client populations.

Table 2: Test result of DeBERTaV3 trained under different client scales.

	Number of Clients				
	8	16	24	32	40
CoLA	61.51	50.21	44.87	35.60	31.26
SST-2	94.30	93.77	93.15	92.30	91.38
MRPC	82.82	77.38	71.44	68.38	64.38
STS-B	86.84	85.90	85.12	80.39	79.35
QQP	88.35	87.32	86.76	86.32	86.00
MNLI	88.20	88.07	87.57	87.36	86.52
QNLI	91.79	91.09	90.45	89.45	89.20
RTE	60.01	59.33	57.16	52.71	50.90
Average	81.73	79.13	77.07	74.06	72.37

5.2.4 Effect of Sampling Methods

We evaluate LEFF employing three distinct sampling strategies: importance-based, round-robin (sequential block fine-tuning per communication round), and random (arbitrary block selection per round). As illustrated in Figure 6 and Figure 7, the importance-based sampling strategy consistently outperforms the round-robin and random strategies across metrics from both the GLUE and E2E NLG benchmarks. On the GLUE benchmark, this performance advantage is particularly pronounced on challenging tasks such as CoLA. Similar trends are observed for MRPC and RTE, although the performance gap diminishes for tasks where all methods achieve high scores. For tasks within the E2E NLG benchmark, importance-based sampling demonstrates consistent superiority, yielding substantial improvements in METEOR and CIDEr scores, alongside modest gains in BLEU and NIST metrics. These results underscore its effectiveness across diverse natural language tasks.

Table 3: Test result of GPT-2 trained under different client scales.

	Number of Clients				
	8	16	24	32	40
BLEU	0.5762	0.5728	0.5726	0.5715	0.5650
NIST	7.4802	7.2763	7.2368	7.1747	6.8036
METEOR	0.3448	0.3401	0.3384	0.3352	0.3275
ROUGE	0.6056	0.6042	0.6033	0.6018	0.5981
CIDEr	1.5174	1.5139	1.5086	1.4988	1.4391
Average	2.1048	2.0615	2.0519	2.0364	1.9467

6 Conclusion

This paper introduced LEFF, a novel federated fine-tuning approach for LLMs that employs selective layer-wise fine-tuning to balance computational efficiency and model performance, achieving a theoretical convergence rate of $\mathcal{O}(1/\sqrt{T})$. Experimental evaluations on the GLUE benchmark and E2E NLG challenge demonstrate LEFF's performance is comparable to full fine-tuning and surpasses other parameter-efficient methods, highlighting its efficacy for resource-constrained, heterogeneous environments. Despite these strengths, LEFF's layer selection and compression introduce an inherent efficiency-fidelity trade-off and an approximation error ($\bar{\Delta}^2$) that can cap performance, as indicated by our theory. Additionally, LEFF increases server-side computational load for tasks like layer importance calculation and specialized aggregation, and can incur substantial server-to-client communication for customized model components, particularly with lighter compression. Future research could mitigate these limitations by developing more sophisticated adaptive compression strategies, optimizing server-side operations, or exploring alternative proxy data utilization methods.

References

Babakniya Sara, Elkordy Ahmed Roushdy, Ezzeldin Yahya H, Liu Qingfeng, Song Kee-Bong, El-Khamy Mostafa, Avestimehr Salman. SLoRA: Federated parameter efficient fine-tuning of language models // arXiv preprint arXiv:2308.06522. 2023.

Bahri Yasaman, Dyer Ethan, Kaplan Jared, Lee Jaehoon, Sharma Utkarsh. Explaining neural scaling laws // Proceedings of the National Academy of Sciences. 2024. 121, 27. e2311878121.

Bai Jiamu, Chen Daoyuan, Qian Bingchen, Yao Liuyi, Li Yaliang. Federated Fine-tuning of Large Language Models under Heterogeneous Tasks and Client Resources // The Thirty-eighth Annual Conference on Neural Information Processing Systems. 2024.

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

Chen Haokun, Zhang Yao, Krompass Denis, Gu Jindong, Tresp Volker. FedDAT: An Approach for Foundation Model Finetuning in Multi-Modal Heterogeneous Federated Learning // Proceedings of the AAAI Conference on Artificial Intelligence. 38, 10. mar 2024. 11285–11293.

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

Cho Yae Jee, Liu Luyang, Xu Zheng, Fahrezi Aldi, Barnes Matt, Joshi Gauri. Heterogeneous LoRA for Federated Fine-tuning of On-device Foundation Models // International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023. 2023.

Fang Zihan, Lin Zheng, Chen Zhe, Chen Xianhao, Gao Yue, Fang Yuguang. Automated federated pipeline for parameter-efficient fine-tuning of large language models // arXiv preprint arXiv:2404.06448. 2024.

Fu Lele, Yang Jinghua, Chen Chuan, Zhang Chuanfu. Low-rank tensor approximation with local structure for multi-view intrinsic subspace clustering // Information Sciences. 2022. 606. 877–891.

Gardent Claire, Shimorina Anastasia, Narayan Shashi, Perez-Beltrachini Laura. Creating Training Corpora for NLG Micro-Planners // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017. 179–188.

Grattafiori Aaron, Dubey Abhimanyu, Jauhri Abhinav, Pandey Abhinav, Kadian Abhishek. The Llama 3 Herd of Models. 2024.

Guo Tao, Guo Song, Wang Junxiao. pFedPrompt: Learning Personalized Prompt for Vision-Language Models in Federated Learning // Proceedings of the ACM Web Conference 2023. 2023. 1364–1374. (WWW '23).

Guo Tao, Guo Song, Wang Junxiao, Tang Xueyang, Xu Wenchao. PromptFL: Let Federated Participants Cooperatively Learn Prompts Instead of Models – Federated Learning in Age of Foundation Model // IEEE Transactions on Mobile Computing. 2024. 23, 5. 5179–5194.

Han Zeyu, Gao Chao, Liu Jinyang, Zhang Sai Qian. Parameter-efficient fine-tuning for large models: A comprehensive survey // arXiv preprint arXiv:2403.14608. 2024.

He Pengcheng, Gao Jianfeng, Chen Weizhu. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing // arXiv preprint arXiv:2111.09543. 2021.

Hendrycks Dan, Burns Collin, Basart Steven, Zou Andy, Mazeika Mantas, Song Dawn, Steinhardt Jacob. Measuring Massive Multitask Language Understanding // International Conference on Learning Representations. 2021.

Houlsby Neil, Giurgiu Andrei, Jastrzebski Stanislaw, Morrone Bruna, De Laroussilhe Quentin, Gesmundo Andrea, Attariyan Mona, Gelly Sylvain. Parameter-Efficient Transfer Learning for NLP // Proceedings of the 36th International Conference on Machine Learning. 97. 2019. 2790–2799. (Proceedings of Machine Learning Research).

Hu Edward J., Shen Yelong, Wallis Phillip, Allen-Zhu Zeyuan, Li Yuanzhi, Wang Shean, Wang Lu, Chen Weizhu. LoRA: Low-Rank Adaptation of Large Language Models // International Conference on Learning Representations. 2022.

Huang Wenke, Ye Mang, Du Bo. Learn From Others and Be Yourself in Heterogeneous Federated Learning // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). jun 2022. 10143–10153.

Huang Wenke, Ye Mang, Shi Zekun, Li He, Du Bo. Rethinking Federated Learning with Domain Shift: A Prototype View // 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023. 16312–16322.

Huang Wenke, Ye Mang, Shi Zekun, Wan Guancheng, Li He, Du Bo, Yang Qiang. Federated Learning for Generalization, Robustness, Fairness: A Survey and Benchmark // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2024. 46, 12. 9387–9406.

Karimireddy Sai Praneeth, Kale Satyen, Mohri Mehryar, Reddi Sashank, Stich Sebastian, Suresh Ananda Theertha. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning // Proceedings of the 37th International Conference on Machine Learning. nov 2020. 5132–5143.

Legate Gwen, Bernier Nicolas, Page-Caccia Lucas, Oyallon Edouard, Belilovsky Eugene. Guiding The Last Layer in Federated Learning with Pre-Trained Models // Advances in Neural Information Processing Systems. 36. 2023. 69832–69848.

Lester Brian, Al-Rfou Rami, Constant Noah. The power of scale for parameter-efficient prompt tuning // arXiv preprint arXiv:2104.08691. 2021.

Li Junyi, Tang Tianyi, Zhao Wayne Xin, Nie Jian-Yun, Wen Ji-Rong. Pre-Trained Language Models for Text Generation: A Survey // ACM Computing Surveys. apr 2024. 56, 9.

Li Xiang, Huang Kaixuan, Yang Wenhao, Wang Shusen, Zhang Zhihua. On the Convergence of FedAvg on Non-IID Data. jun 2020.

Loshchilov Ilya, Hutter Frank. Decoupled Weight Decay Regularization. jan 2019.

McMahan Brendan, Moore Eider, Ramage Daniel, Hampson Seth, Arcas Blaise Aguera y. Communication-Efficient Learning of Deep Networks from Decentralized Data // Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. 54. 2017. 1273–1282. (Proceedings of Machine Learning Research).

Molchanov Pavlo, Mallya Arun, Tyree Stephen, Frosio Iuri, Kautz Jan. Importance Estimation for Neural Network Pruning // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). jun 2019.

Novikova Jekaterina, Dušek Ondrej, Rieser Verena. The E2E Dataset: New Challenges for End-to-End Generation // Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue. 2017. arXiv:1706.09254.

Pan Rui, Liu Xiang, Diao Shizhe, Pi Renjie, Zhang Jipeng, Han Chi, Zhang Tong. LISA: Layerwise Importance Sampling for Memory-Efficient Large Language Model Fine-Tuning // arXiv preprint arXiv:2403.17919. 2024.

Qi Zhuang, Meng Lei, Chen Zitan, Hu Han, Lin Hui, Meng Xiangxu. Cross-silo prototypical calibration for federated learning with non-iid data // Proceedings of the 31st ACM International Conference on Multimedia. 2023. 3099–3107.

Qin Zhen, Chen Daoyuan, Qian Bingchen, Ding Bolin, Li Yaliang, Deng Shuiguang. Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes // Proceedings of the 41st International Conference on Machine Learning. 2024. (ICML'24).

Radford Alec, Wu Jeffrey, Child Rewon, Luan David, Amodei Dario, Sutskever Ilya. Language models are unsupervised multitask learners // OpenAI blog. 2019. 1, 8. 9.

Ranathunga Surangika, Lee En-Shiun Annie, Prifti Skenduli Marjana, Shekhar Ravi, Alam Mehreen, Kaur Rishemjit. Neural Machine Translation for Low-resource Languages: A Survey // ACM Computing Surveys. feb 2023. 55, 11.

Shen Jinglong, Cheng Nan, Wang Xiucheng, Lyu Feng, Xu Wenchao, Liu Zhi, Aldubaikhy Khalid, Shen Xuemin. RingSFL: An Adaptive Split Federated Learning Towards Taming Client Heterogeneity // IEEE Transactions on Mobile Computing. 2024. 23, 5. 5462–5478.

Shu Yao, Hu Wenyang, Ng See-Kiong, Low Bryan Kian Hsiang, Yu Fei Richard. Ferret: Federated Full-Parameter Tuning at Scale for Large Language Models. 2025.

Sun Guangyu, Mendieta Matias, Yang Taojiannan, Chen Chen. Conquering the communication constraints to enable large pre-trained models in federated learning // arXiv preprint arXiv:2210.01708. 2022.

Wang Alex, Singh Amanpreet, Michael Julian, Hill Felix, Levy Omer, Bowman Samuel R. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding // International Conference on Learning Representations. 2019.

Wang Haozhao, Jia Yabo, Zhang Meng, Hu Qinghao, Ren Hao, Sun Peng, Wen Yonggang, Zhang Tianwei. FedDSE: Distribution-aware Sub-model Extraction for Federated Learning over Resource-constrained Devices // Proceedings of the ACM Web Conference 2024. 2024a. 2902–2913. (WWW '24).

Wang Haozhao, Zheng Peirong, Han Xingshuo, Xu Wenchao, Li Ruixuan, Zhang Tianwei. FedNLR: Federated Learning with Neuron-wise Learning Rates // Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024b. 3069–3080.

Wang Lin, Wang Zhichao, Tang Xiaoying. Save It All: Enabling Full Parameter Tuning for Federated Large Language Models via Cycle Block Gradient Descent. 2024c.

Wang Ziyao, Shen Zheyu, He Yexiao, Sun Guoheng, Wang Hongyi, Lyu Lingjuan, Li Ang. FLoRA: Federated Fine-Tuning Large Language Models with Heterogeneous Low-Rank Adaptations // The Thirty-eighth Annual Conference on Neural Information Processing Systems. 2024d.

Woisetschläger Herbert, Erben Alexander, Wang Shiqiang, Mayer Ruben, Jacobsen Hans-Arno. Federated Fine-Tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly // Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning. 2024. 39–50. (DEEM '24).

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

Yu Fei, Zhang Hongbo, Tiwari Prayag, Wang Benyou. Natural Language Reasoning, A Survey // ACM Computing Surveys. may 2024. Just Accepted.

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

Zaken Elad Ben, Ravfogel Shauli, Goldberg Yoav. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models // arXiv preprint arXiv:2106.10199. 2021.

Zhang Jianyi, Vahidian Saeed, Kuo Martin, Li Chunyuan, Zhang Ruiyi, Yu Tong, Wang Guoyin, Chen Yiran. Towards Building The FederatedGPT: Federated Instruction Tuning // ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2024. 6915–6919.

Zhang Zhuo, Yang Yuanhang, Dai Yong, Qifan Wang, Yu Yue, Qu Lizhen, Xu Zenglin. FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models // Findings of the Association for Computational Linguistics: ACL 2023. 2023. 9963–9977.

Zhuang Weiming, Chen Chen, Lyu Lingjuan. When foundation model meets federated learning: Motivations, challenges, and future directions // arXiv preprint arXiv:2306.15546. 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the LEFF framework, its mechanisms for efficiency and performance preservation (e.g., layer selection, model compression), its handling of data and system heterogeneity, the theoretical convergence rate claim, and the claim of superior empirical results. These aspects are subsequently detailed and supported in Sections 3 (Methodology), 4 (Convergence Analysis), and 5 (Evaluations).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Conclusion (Section 6) explicitly discusses limitations, including the efficiency-fidelity trade-off due to layer selection and compression, the impact of approximation error ($\bar{\Delta}^2$) on performance, increased server-side computational load, and potential server-to-client communication overhead.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 4 (specifically Subsection 4.1) lists the assumptions (Assumptions 1-4) for the theoretical analysis. The main theoretical results (Theorem 1 and Corollary 1) are presented, and the paper states that proofs are provided in the Appendix (Sections A, B).

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 (specifically Subsection 5.1) details the experimental setup, including baselines, models (GPT-2 Medium, DeBERTaV3 Base), datasets (GLUE, E2E NLG, WebNLG, WNLI), data heterogeneity settings (Dirichlet α), number of clients, communication rounds, and local epochs. Key components of LEFF like layer sampling and model compression are described in Section 3. The provided information covers the core aspects needed to understand and attempt reproduction.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper uses public datasets (GLUE, E2E NLG, WebNLG, WNLI) and existing pre-trained models (GPT-2, DeBERTaV3), which are cited. However, it does not explicitly state that the code for the proposed LEFF method or the experimental scripts are publicly available, nor does it provide a link or instructions for accessing them.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.1 (Experimental Setup) specifies the models (GPT-2 Medium, DeBERTaV3 Base), datasets (GLUE, E2E NLG, WebNLG, WNLI), data partitioning (Dirichlet with α), optimizer (AdamW), learning rate (1×10^{-5}), communication rounds (20), and local epochs (1). Key aspects of LEFF like layer sampling and model compression (including the analysis of compression ratio r) are described.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The authors provide the standard deviation of the experimental results in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5.1 (Experimental Setup) states that 'All experiments were performed on a system with eight NVIDIA H100 GPUs.' This specifies the type and number of compute workers.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research focuses on algorithmic development for efficient federated fine-tuning of language models using publicly available datasets. The work aims to improve efficiency and manage resource constraints, with no apparent direct ethical concerns or violations of the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: A discussion of broader impacts is provided in the appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper proposes a new fine-tuning method (LEFF) and evaluates it using existing pre-trained models (GPT-2, DeBERTaV3) and public datasets. It does not introduce or release new models or datasets that would inherently pose a high risk for misuse requiring specific safeguards developed by the authors.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly cites the original sources for existing assets like pre-trained models (GPT-2 Radford et al. (2019), DeBERTaV3 He et al. (2021)) and datasets (GLUE Wang et al. (2019), E2E NLG Novikova et al. (2017), WebNLG Gardent et al. (2017)).

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper introduces a new method (LEFF). The paper itself serves as documentation for this method. No new datasets or standalone models are introduced for release as distinct assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The research does not involve crowdsourcing experiments or new research with human subjects; it utilizes existing public datasets.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The research does not involve new studies with human subjects, so IRB approval or discussion of participant risks is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The research focuses on developing a new method (LEFF) for fine-tuning LLMs. LLMs are the subject of the study (e.g., GPT-2, DeBERTaV3 are fine-tuned), not a tool used in an important, original, or non-standard way to develop the core LEFF methodology itself (e.g., for algorithm design or proof generation).

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Proof of Lemma 1

A.1 Bound on the Expected Local Gradient Norm

Proof. We use the standard variance decomposition $\mathbb{E}[\|X\|^2] = \mathbb{E}[\|X - \mathbb{E}[X]\|^2] + \|\mathbb{E}[X]\|^2$.

$$\mathbb{E}_\xi \left[\|G_{i,t,k}\|^2 \right] = \mathbb{E}_\xi \left[\|G_{i,t,k} - \mathbb{E}_\xi[G_{i,t,k}]\|^2 \right] + \|\mathbb{E}_\xi[G_{i,t,k}]\|^2 . \tag{19}$$

By definition, $\mathbb{E}_\xi[G_{i,t,k}] = \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta^{\mathcal{L}_i}_{i,t,k} | \check{\Theta}^{\mathcal{L}_i^-}_{g,t})$. The first term is the variance of the stochastic gradient given the current local model $\Theta^{\mathcal{L}_i}_{i,t,k}$ and the (potentially compressed) non-updated parameters $\check{\Theta}^{\mathcal{L}_i^-}_{g,t}$. Assumption 2 provides a bound on the variance of the gradient for the full model: $\mathbb{E}_\xi[\|\nabla \mathcal{F}_i(\Theta; \xi) - \nabla \mathcal{F}_i(\Theta)\|^2] \leq \sigma_i^2 \leq \sigma^2$. We assume this bound also applies to the variance of the stochastic gradient computed only for the selected layers $\mathcal{L}_i$:

$$\mathbb{E}_\xi \left[\|G_{i,t,k} - \mathbb{E}_\xi[G_{i,t,k}]\|^2 \right] \leq \sigma^2 \tag{20}$$

For the second term, the squared norm of the expected gradient, we use the inequality $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$:

$$\begin{aligned} \|\mathbb{E}_\xi[G_{i,t,k}]\|^2 &= \left\| \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta^{\mathcal{L}_i}_{i,t,k} | \check{\Theta}^{\mathcal{L}_i^-}_{g,t}) \right\|^2 \\ &= \left\| \left(\nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta^{\mathcal{L}_i}_{i,t,k} | \check{\Theta}^{\mathcal{L}_i^-}_{g,t}) - \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t}) \right) + \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t}) \right\|^2 \\ &\leq 2 \left\| \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta^{\mathcal{L}_i}_{i,t,k} | \check{\Theta}^{\mathcal{L}_i^-}_{g,t}) - \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t}) \right\|^2 + 2 \left\| \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t}) \right\|^2 \end{aligned} \tag{21}$$

The first term in (21) captures the deviation of the expected local gradient at step k from the true local gradient (for layers $\mathcal{L}_i$) evaluated at the round's initial global model $\Theta_{g,t}$. This deviation arises from both the drift of the local model ($\Theta^{\mathcal{L}_i}_{i,t,k}$ vs $\Theta^{\mathcal{L}_i}_{g,t}$) and the use of potentially approximated parameters ($\check{\Theta}^{\mathcal{L}_i^-}_{g,t}$). Assumption 4 bounds the average deviation over K steps: $\mathbb{E}_{\mathcal{L}_i \sim \mathbf{p}_i}[\|\bar{G}_{i,t} - \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t})\|^2] \leq \Delta^2_{i,t}$, where $\bar{G}_{i,t} = (1/K) \sum_{k=0}^{K-1} \mathbb{E}_\xi[G_{i,t,k}]$. While Assumption 4 applies to the average, analyses often rely on bounding the instantaneous deviation. We make a simplifying assumption, that the expected instantaneous deviation (over $\mathcal{L}_i$) is also related to the average approximation error bound Δ_t^2. Specifically, we assume:

$$\mathbb{E}_{\mathcal{L}_i \sim \mathbf{p}_i} \left[\left\| \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta^{\mathcal{L}_i}_{i,t,k} | \check{\Theta}^{\mathcal{L}_i^-}_{g,t}) - \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t}) \right\|^2 \right] \leq c_a \Delta_t^2 \tag{22}$$

for some constant $c_a \geq 1$.

Combining (20), (21), and (22), and taking expectation over $\mathcal{L}_i \sim \mathbf{p}_i$:

$$\begin{aligned} &\mathbb{E}_{\mathcal{L}_i} \mathbb{E}_\xi \left[\|G_{i,t,k}\|^2 \right] \\ &\leq \mathbb{E}_{\mathcal{L}_i} \left[\sigma^2 + \|\mathbb{E}_\xi[G_{i,t,k}]\|^2 \right] \\ &\leq \sigma^2 + \mathbb{E}_{\mathcal{L}_i} \left[2 \left\| \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta^{\mathcal{L}_i}_{i,t,k} | \check{\Theta}^{L_i^-}_{g,t}) - \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t}) \right\|^2 + 2 \left\| \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t}) \right\|^2 \right] \\ &\leq \sigma^2 + 2c_a \Delta_t^2 + 2\mathbb{E}_{\mathcal{L}_i} \left[\left\| \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t}) \right\|^2 \right] \\ &\leq c_b \left(\sigma^2 + \Delta_t^2 + \mathbb{E}_{\mathcal{L}_i} \left[\left\| \nabla_{\Theta^{\mathcal{L}_i}} \mathcal{F}_i(\Theta_{g,t}) \right\|^2 \right] \right) \end{aligned} \tag{23}$$

where $c_b = 2c_a$ is a constant. □

A.2 Bound on the Global Model Update Norm

Proof. The squared norm of the global model change is the sum of the squared norms of the changes in each layer:

$$\mathbb{E} \left[\|\Theta_{g,t+1} - \Theta_{g,t}\|^2 \right] = \mathbb{E} \left[\sum_{l=1}^{L} \left\| \Theta^l_{g,t+1} - \Theta^l_{g,t} \right\|^2 \right] = \sum_{l=1}^{L} \mathbb{E} \left[\left\| \Theta^l_{g,t+1} - \Theta^l_{g,t} \right\|^2 \right], \tag{24}$$

where $\Theta^l_{g,t}$ denotes the parameters of the l-th layer of the global model at round t.

The global model update for layer l at round $t+1$ is given by the weighted average of the corresponding layer parameters from clients that updated this layer:

$$\Theta^l_{g,t+1} = \sum_{i \in \mathcal{S}^l_t} w^l_i \Theta^l_{i,t,K}, \tag{25}$$

where $\mathcal{S}^l_t = \{i \in \mathcal{A}_t \mid l \in \mathcal{L}_i\}$ is the set of clients selected in round t ($\mathcal{A}_t$) that included layer l in their layer block $\mathcal{L}_i$, $w^l_i = D_i / (\sum_{j \in \mathcal{S}^l_t} D_j)$ are the aggregation weights (with $\sum_{i \in \mathcal{S}^l_t} w^l_i = 1$, $w^l_i \geq 0$), and $\Theta^l_{i,t,K}$ is the l-th layer parameter of client i after K local steps starting from $\Theta^l_{i,t,0} = \Theta^l_{g,t}$.

The local update process for layer l on client i (if $l \in \mathcal{L}_i$) follows:

$$\Theta^l_{i,t,K} = \Theta^l_{i,t,0} - \eta \sum_{k=0}^{K-1} G^l_{i,t,k} = \Theta^l_{g,t} - \eta \sum_{k=0}^{K-1} G^l_{i,t,k}, \tag{26}$$

where $G^l_{i,t,k}$ is the component of the stochastic gradient $G_{i,t,k}$ (computed by client i at local step k using data sample $\xi_{i,t,k}$) corresponding to layer l.

Therefore, the change in the global model's l-th layer is:

$$\Theta^l_{g,t+1} - \Theta^l_{g,t} = \sum_{i \in \mathcal{S}^l_t} w^l_i (\Theta^l_{i,t,K} - \Theta^l_{g,t}) = -\eta \sum_{i \in S^l_t} w^l_i \left(\sum_{k=0}^{K-1} G^l_{i,t,k} \right). \tag{27}$$

We want to bound the expected squared norm $\mathbb{E}[||\Theta^l_{g,t+1} - \Theta^l_{g,t}||^2]$. Let $\delta^l_{i,t} = \Theta^l_{i,t,K} - \Theta^l_{g,t} = -\eta \sum_{k=0}^{K-1} G^l_{i,t,k}$ denote the total update applied by client i to layer l (if $l \in \mathcal{L}_i$).

$$\mathbb{E}\left[\left\| \Theta^l_{g,t+1} - \Theta^l_{g,t} \right\|^2 \right] = \mathbb{E}\left[\left\| \sum_{i \in \mathcal{S}^l_t} w^l_i \delta^l_{i,t} \right\|^2 \right]. \tag{28}$$

The expectation $\mathbb{E}[\cdot]$ is over all sources of randomness: $\mathcal{A}_t$, $\{\mathcal{L}_i\}_{i \in \mathcal{A}_t}$, and $\{\xi_{i,t,k}\}_{i \in \mathcal{A}_t, 0 \leq k < K}$.

Since $\| \cdot \|^2$ is a convex function and $\sum_{i \in \mathcal{S}^l_t} w^l_i = 1$ with $w^l_i \geq 0$, we can apply Jensen's inequality:

$$\left\| \sum_{i \in S^l_t} w^l_i \delta^l_{i,t} \right\|^2 \leq \sum_{i \in \mathcal{S}^l_t} w^l_i \left\| \delta^l_{i,t} \right\|^2. \tag{29}$$

Taking the expectation over all randomness:

$$\mathbb{E}\left[\left\| \Theta^l_{g,t+1} - \Theta^l_{g,t} \right\|^2 \right] \leq \mathbb{E}\left[\sum_{i \in \mathcal{S}^l_t} w^l_i \left\| \delta^l_{i,t} \right\|^2 \right]. \tag{30}$$

We can rewrite the expectation using the law of total expectation, conditioning first on the client and layer selections ($\mathcal{A}_t, \mathcal{L}_i$), which determines the set $\mathcal{S}^l_t$ and weights w^l_i:

$$\mathbb{E}\left[\left\| \Theta^l_{g,t+1} - \Theta^l_{g,t} \right\|^2 \right] \leq \mathbb{E}_{\mathcal{A}_t, \mathcal{L}_i}\left[\sum_{i \in \mathcal{S}^l_t} w^l_i \mathbb{E}_\xi \left[\left\| \delta^l_{i,t} \right\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right] \right], \tag{31}$$

where $\mathbb{E}_\xi[\cdot \mid \mathcal{A}_t, \mathcal{L}_i]$ denotes the expectation over the stochasticity of local gradients $\{\xi_{i,t,k}\}$ given the client and layer selections. Note that the condition $i \in \mathcal{S}^l_t$ implies $i \in \mathcal{A}_t$ and $l \in \mathcal{L}_i$.

Now, we bound the inner term $\mathbb{E}_\xi[\|\delta^l_{i,t}\|^2 \mid \mathcal{A}_t, \mathcal{L}_i]$ for $i \in \mathcal{S}^l_t$:

$$\mathbb{E}_\xi\left[\left\| \delta^l_{i,t} \right\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right] = \mathbb{E}_\xi\left[\left\| -\eta \sum_{k=0}^{K-1} G^l_{i,t,k} \right\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right] = \eta^2 \mathbb{E}_\xi\left[\left\| \sum_{k=0}^{K-1} G^l_{i,t,k} \right\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right]. \tag{32}$$

Using the standard bound $\mathbb{E}[||\sum_{k=0}^{K-1} X_k||^2] \leq K \sum_{k=0}^{K-1} \mathbb{E}[||X_k||^2]$ (often used in FL analysis Karimireddy et al. (2020)):

$$\mathbb{E}_\xi \left[\left\| \sum_{k=0}^{K-1} G^l_{i,t,k} \right\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right] \leq K \sum_{k=0}^{K-1} \mathbb{E}_\xi \left[\left\| G^l_{i,t,k} \right\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right]. \tag{33}$$

Since $G^l_{i,t,k}$ is the l-th layer component of the full gradient $G_{i,t,k}$, we have $\|G^l_{i,t,k}\|^2 \leq \|G_{i,t,k}\|^2$. Thus:

$$\mathbb{E}_\xi \left[\left\| G^l_{i,t,k} \right\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right] \leq \mathbb{E}_\xi \left[\|G_{i,t,k}\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right]. \tag{34}$$

From Assumptions 1, 2, and 4, the expected squared norm of the stochastic gradient $G_{i,t,k}$ computed by client i at local step k can be bounded. A common bound derived under these assumptions in (23) takes the form:

$$\mathbb{E}_\xi \left[\|G_{i,t,k}\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right] \leq C_{\text{grad}} \left(\sigma^2 + \Delta_t^2 + \|\nabla \mathcal{F}_i(\Theta_{g,t})\|^2 \right), \tag{35}$$

where C_{grad} is a constant, and we use $\Theta_{g,t}$ as a reference point, absorbing dependencies on the intermediate local models $\Theta_{i,t,k}$ into the constant C_{grad} (this is a common simplification in FL analysis, valid for sufficiently small ηK). Let $X_{i,t} = \sigma^2 + \Delta_t^2 + \|\nabla \mathcal{F}_i(\Theta_{g,t})\|^2$. Then:

$$\mathbb{E}_\xi \left[\left\| G^l_{i,t,k} \right\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right] \leq C_{\text{grad}} X_{i,t}. \tag{36}$$

Substituting back:

$$\mathbb{E}_\xi \left[\left\| \delta^l_{i,t} \right\|^2 \mid \mathcal{A}_t, \mathcal{L}_i \right] \leq \eta^2 K \sum_{k=0}^{K-1} (C_{\text{grad}} X_{i,t}) = \eta^2 K^2 C_{\text{grad}} X_{i,t}. \tag{37}$$

Now, substitute this into the bound for $\mathbb{E}[||\Theta^l_{g,t+1} - \Theta^l_{g,t}||^2]$:

$$\begin{aligned} \mathbb{E} \left[\left\| \Theta^l_{g,t+1} - \Theta^l_{g,t} \right\|^2 \right] &\leq \mathbb{E}_{\mathcal{A}_t, \mathcal{L}_i} \left[\sum_{i \in \mathcal{S}^l_t} w^l_i (\eta^2 K^2 C_{\text{grad}} X_{i,t}) \right] \\ &= \eta^2 K^2 C_{\text{grad}} \mathbb{E}_{\mathcal{A}_t, \mathcal{L}_i} \left[\sum_{i \in \mathcal{S}^l_t} w^l_i \left(\sigma^2 + \Delta_t^2 + \|\nabla \mathcal{F}_i(\Theta_{g,t})\|^2 \right) \right] \end{aligned} \tag{38}$$

Let $\mathbb{E}_{\mathcal{S}^l_t}[\cdot]$ denote the expectation $\mathbb{E}_{\mathcal{A}_t, \mathcal{L}_i}[\cdot]$. The term $\mathbb{E}_{\mathcal{S}^l_t}[\sum_{i \in \mathcal{S}^l_t} w^l_i(\cdot)]$ represents the expected weighted average over clients participating in the update for layer l. Under assumptions of unbiased client sampling (e.g., uniform random sampling of $\mathcal{A}_t$ clients) and potentially layer sampling ($\mathcal{L}_i$), this expectation can be related to the average over all clients. Assuming there exists a constant c_d (which may depend on the sampling strategy, e.g., A/N, and the distribution of weights w^l_i) such that:

$$\mathbb{E}_{\mathcal{S}^l_t} \left[\sum_{i \in \mathcal{S}^l_t} w^l_i X_{i,t} \right] \leq c_d \frac{1}{N} \sum_{i=1}^{N} X_{i,t}. \tag{39}$$

Then:

$$\begin{aligned} \mathbb{E} \left[\left\| \Theta^l_{g,t+1} - \Theta^l_{g,t} \right\|^2 \right] &\leq \eta^2 K^2 C_{\text{grad}} c_d \frac{1}{N} \sum_{i=1}^{N} \left(\sigma^2 + \Delta_t^2 + \|\nabla \mathcal{F}_i(\Theta_{g,t})\|^2 \right) \\ &= \eta^2 K^2 C_{\text{grad}} c_d \left(\sigma^2 + \Delta_t^2 + \frac{1}{N} \sum_{i=1}^{N} \|\nabla \mathcal{F}_i(\Theta_{g,t})\|^2 \right). \end{aligned} \tag{40}$$

Now we use Assumption 3 (Bounded Heterogeneity): $\frac{1}{N}\sum_{i=1}^{N}||\nabla\mathcal{F}_i(\Theta)-\nabla\mathcal{F}(\Theta)||^2\leq\zeta^2$. We have the decomposition:

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\|\nabla\mathcal{F}_i(\Theta_{g,t})\|^2=\frac{1}{N}\sum_{i=1}^{N}\|\nabla\mathcal{F}_i(\Theta_{g,t})-\nabla\mathcal{F}(\Theta_{g,t})+\nabla\mathcal{F}(\Theta_{g,t})\|^2\\
&=\frac{1}{N}\sum_{i=1}^{N}\left(\|\nabla\mathcal{F}_i(\Theta_{g,t})-\nabla\mathcal{F}(\Theta_{g,t})\|^2+\|\nabla\mathcal{F}(\Theta_t^g)\|^2+2\langle\nabla\mathcal{F}_i(\Theta_{g,t})-\nabla\mathcal{F}(\Theta_{g,t}),\nabla\mathcal{F}(\Theta_{g,t})\rangle\right)\\
&=\left(\frac{1}{N}\sum_{i=1}^{N}\|\nabla\mathcal{F}_i(\Theta_{g,t})-\nabla\mathcal{F}(\Theta_{g,t})\|^2\right)+\|\nabla\mathcal{F}(\Theta_{g,t})\|^2\\
&\quad+2\left\langle\frac{1}{N}\sum_{i=1}^{N}\nabla\mathcal{F}_i(\Theta_{g,t})-\nabla\mathcal{F}(\Theta_{g,t}),\nabla\mathcal{F}(\Theta_{g,t})\right\rangle.
\end{aligned}\tag{41}$$

Since $\nabla\mathcal{F}(\Theta_{g,t})=\frac{1}{N}\sum_{i=1}^{N}\nabla\mathcal{F}_i(\Theta_{g,t})$, the cross term is zero. Using Assumption 3:

$$\frac{1}{N}\sum_{i=1}^{N}\|\nabla\mathcal{F}_i(\Theta_{g,t})\|^2\leq\zeta^2+\|\nabla\mathcal{F}(\Theta_{g,t})\|^2.\tag{42}$$

Substituting this into the bound for the layer update:

$$\mathbb{E}\left[\left\|\Theta_{g,t+1}^l-\Theta_{g,t}^l\right\|^2\right]\leq\eta^2K^2C_{\text{grad}}c_d\left(\sigma^2+\Delta_t^2+\zeta^2+\|\nabla\mathcal{F}(\Theta_{g,t})\|^2\right).\tag{43}$$

Finally, sum over all layers $l=1,\ldots,L$:

$$\begin{aligned}
\mathbb{E}\left[\|\Theta_{g,t+1}-\Theta_{g,t}\|^2\right]&=\sum_{l=1}^{L}\mathbb{E}\left[\left\|\Theta_{g,t+1}^l-\Theta_{g,t}^l\right\|^2\right]\\
&\leq\sum_{l=1}^{L}\left[\eta^2K^2C_{\text{grad}}c_d\left(\sigma^2+\Delta_t^2+\zeta^2+||\nabla\mathcal{F}(\Theta_{g,t})||^2\right)\right]\\
&=L\eta^2K^2C_{\text{grad}}c_d\left(\sigma^2+\Delta_t^2+\zeta^2+||\nabla\mathcal{F}(\Theta_{g,t})||^2\right)
\end{aligned}\tag{44}$$

Let $C_3=LC_{\text{grad}}c_d$. This constant incorporates the number of layers L, the constant from the gradient bound C_{grad}, and the constant related to sampling and weighting c_d. This yields the final result:

$$\mathbb{E}\left[\|\Theta_{g,t+1}-\Theta_{g,t}\|^2\right]\leq C_3\eta^2K^2\left(\sigma^2+\Delta_t^2+\zeta^2+\|\nabla\mathcal{F}(\Theta_{g,t})\|^2\right).\tag{45}$$

This completes the proof. □

B Proof of Main Theorem

Proof. In this section, we present the convergence analysis for our algorithm. Our goal is to bound the average squared norm of the global gradient over T communication rounds.

We begin with the L-smoothness property of the global objective function $\mathcal{F}$, stated in **Assumption 1**. By the definition of L-smoothness, often referred to as the Descent Lemma or Quadratic Upper Bound, we have:

$$\mathbb{E}[\mathcal{F}(\Theta_{g,t+1})]\leq\mathcal{F}(\Theta_{g,t})+\mathbb{E}[\langle\nabla\mathcal{F}(\Theta_{g,t}),\Theta_{g,t+1}-\Theta_{g,t}\rangle]+\frac{L}{2}\mathbb{E}[\|\Theta_{g,t+1}-\Theta_{g,t}\|^2],\tag{46}$$

where the expectation $\mathbb{E}[\cdot]$ is taken over all sources of randomness up to round $t+1$, including client sampling ($\mathcal{A}_t$), layer sampling ($\mathcal{L}_i$ for $i\in\mathcal{A}_t$), and local stochastic gradient noise ($\xi_{i,t,k}$).

The core of the analysis involves carefully bounding the inner product term $\mathbb{E}[\langle\nabla\mathcal{F}(\Theta_{g,t}),\Theta_{g,t+1}-\Theta_{g,t}\rangle]$ and the quadratic term $\mathbb{E}[\|\Theta_{g,t+1}-\Theta_{g,t}\|^2]$. The bound for the quadratic term is typically established in a separate lemma (referred to as **Lemma 1**). We focus on deriving the bound for the inner product term.

B.1 Bounding the Inner Product Term

Recall the global model update is constructed layer-wise:

$$\Theta_{g,t+1} - \Theta_{g,t} = \{\Theta^l_{g,t+1} - \Theta^l_{g,t}\}_{l=1\dots L} \tag{47}$$

where for each layer l:

$$\Theta^l_{g,t+1} - \Theta^l_{g,t} = -\eta \sum_{i \in \mathcal{S}^l_t} w^l_i \sum_{k=0}^{K-1} G^l_{i,t,k} \tag{48}$$

Here, $\mathcal{S}^l_t = \{i \in \mathcal{A}_t \mid l \in \mathcal{L}_i\}$ is the set of selected clients that update layer l, w^l_i is the aggregation weight for client i on layer l, and $G^l_{i,t,k}$ is the stochastic gradient computed by client i for layer l at local step k.

The inner product term can be expanded as a sum over layers:

$$\begin{aligned}
\mathbb{E}[\langle \nabla \mathcal{F}(\Theta_{g,t}), \Theta_{g,t+1} - \Theta_{g,t} \rangle] &= \mathbb{E}\left[\sum_{l=1}^{L} \langle \nabla_{\Theta^l} \mathcal{F}(\Theta_{g,t}), \Theta^l_{g,t+1} - \Theta^l_{g,t} \rangle \right] \\
&= \sum_{l=1}^{L} \mathbb{E}\left[\langle \nabla_{\Theta^l} \mathcal{F}(\Theta_{g,t}), -\eta \sum_{i \in \mathcal{S}^l_t} w^l_i \sum_{k=0}^{K-1} G^l_{i,t,k} \rangle \right] \\
&= -\sum_{l=1}^{L} \eta \mathbb{E}\left[\langle \nabla_{\Theta^l} \mathcal{F}(\Theta_{g,t}), \sum_{i \in \mathcal{S}^l_t} w^l_i \sum_{k=0}^{K-1} G^l_{i,t,k} \rangle \right]
\end{aligned} \tag{49}$$

Let $\mathbb{E}_\xi[\cdot]$ denote the expectation over the local data samples ξ, conditioned on the client selection $\mathcal{A}_t$ and layer selections $\mathcal{L}_i$. Let $\bar{G}^l_{i,t} = \frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E}_\xi[G^l_{i,t,k} | \mathcal{A}_t, \mathcal{L}_i]$ be the average expected local gradient for layer l computed by client i (if $l \in \mathcal{L}_i$). Then $\sum_{k=0}^{K-1} \mathbb{E}_\xi[G^l_{i,t,k} | \mathcal{A}_t, \mathcal{L}_i] = K \bar{G}^l_{i,t}$. Taking the expectation over ξ first, we get:

$$\begin{aligned}
\mathbb{E}[\langle \cdots \rangle] &= -\sum_{l=1}^{L} \eta \mathbb{E}_{\mathcal{A}_t, \mathcal{L}_i}\left[\langle \nabla_{\Theta^l} \mathcal{F}(\Theta_{g,t}), \sum_{i \in \mathcal{S}^l_t} w^l_i K \bar{G}^l_{i,t} \rangle \right] \\
&= -K \sum_{l=1}^{L} \eta \mathbb{E}\left[\langle \nabla_{\Theta^l} \mathcal{F}(\Theta_{g,t}), \sum_{i \in \mathcal{S}^l_t} w^l_i \bar{G}^l_{i,t} \rangle \right]
\end{aligned} \tag{50}$$

where the outer expectation $\mathbb{E}[\cdot]$ is now over $\mathcal{A}_t$ and $\mathcal{L}_i$.

Standard techniques in FL analysis (e.g., using algebraic identities like $2\langle a, b \rangle = \|a\|^2 + \|b\|^2 - \|a - b\|^2$ or Young's inequality, and carefully handling the expectations over sampling) allow us to bound the inner product. This involves decomposing the error terms and applying the assumptions. While the detailed derivation is intricate due to the layer selection and weighting scheme, it leads to a bound of the following form (similar to analyses of FedAvg variants):

$$\mathbb{E}[\langle \nabla \mathcal{F}(\Theta_{g,t}), \Theta_{g,t+1} - \Theta_{g,t} \rangle] \leq -C_4 \eta K \|\nabla \mathcal{F}(\Theta_{g,t})\|^2 + C_5 \eta K (\bar{\Delta}^2_t + \zeta^2 + \sigma^2/K) \tag{51}$$

where $\bar{\Delta}^2_t = \frac{1}{N} \sum_i \Delta^2_{i,t}$ is the average model approximation error bound across clients at round t. The constants C_4 and C_5 depend on factors like the client sampling ratio (A/N), layer sampling probabilities ($\mathbf{p}_i$), and potentially the distribution of weights w^l_i. The term $-C_4 \eta K \|\nabla \mathcal{F}(\Theta_{g,t})\|^2$ represents the desired descent along the negative gradient direction, while the term $C_5 \eta K (\bar{\Delta}^2_t + \zeta^2 + \sigma^2/K)$ captures the accumulated error from model approximation, heterogeneity, and residual stochastic gradient variance (scaled by $1/K$ due to averaging K steps implicitly or explicitly in the derivation).

B.2 Completing the Proof

We now substitute the inner product bound (51) and the bound on the quadratic term from **Lemma 1** into the L-smoothness inequality (46). Lemma 1 typically provides a bound like:

$$\mathbb{E}[\|\Theta_{g,t+1} - \Theta_{g,t}\|^2] \leq C_3 \eta^2 K^2 (\sigma^2 + \bar{\Delta}^2_t + \zeta^2 + \|\nabla \mathcal{F}(\Theta_{g,t})\|^2) \tag{52}$$

where C_3 is another constant derived in Lemma 1.

Substituting (51) and (52) into (46):

$$\begin{aligned}\mathbb{E}[\mathcal{F}(\Theta_{g,t+1})] \leq \mathcal{F}(\Theta_{g,t}) - C_4\eta K\|\nabla\mathcal{F}(\Theta_{g,t})\|^2 + C_5\eta K(\bar{\Delta}_t^2 + \zeta^2 + \sigma^2/K) \\ + \frac{L}{2}C_3\eta^2K^2(\sigma^2 + \bar{\Delta}_t^2 + \zeta^2 + \|\nabla\mathcal{F}(\Theta_{g,t})\|^2)\end{aligned} \tag{53}$$

Rearranging the terms to group the gradient norm:

$$\begin{aligned}\mathbb{E}[\mathcal{F}(\Theta_{g,t+1})] \leq \mathcal{F}(\Theta_{g,t}) - \left(C_4\eta K - \frac{LC_3}{2}\eta^2K^2\right)\|\nabla\mathcal{F}(\Theta_{g,t})\|^2 \\ + C_5\eta K(\bar{\Delta}_t^2 + \zeta^2 + \sigma^2/K) + \frac{LC_3}{2}\eta^2K^2(\sigma^2 + \bar{\Delta}_t^2 + \zeta^2)\end{aligned} \tag{54}$$

To ensure convergence, we require the coefficient of the gradient norm term to be positive. We choose the local learning rate η sufficiently small such that $C_4\eta K - \frac{LC_3}{2}\eta^2K^2 > 0$. A standard choice is to make the second term at most half of the first term, which holds if $\eta \leq \frac{C_4}{LC_3K}$. Under this condition:

$$C_4\eta K - \frac{LC_3}{2}\eta^2K^2 \geq C_4\eta K - \frac{LC_3}{2}\left(\frac{C_4}{LC_3K}\right)\eta K = C_4\eta K - \frac{C_4}{2}\eta K = \frac{C_4}{2}\eta K \tag{55}$$

Thus, the inequality becomes:

$$\begin{aligned}\mathbb{E}[\mathcal{F}(\Theta_{g,t+1})] \leq \mathcal{F}(\Theta_{g,t}) - \frac{C_4}{2}\eta K\|\nabla\mathcal{F}(\Theta_{g,t})\|^2 \\ + C_6\eta K(\bar{\Delta}_t^2 + \zeta^2 + \sigma^2/K) + C_7\eta^2K^2(\sigma^2 + \bar{\Delta}_t^2 + \zeta^2)\end{aligned} \tag{56}$$

where we define $C_6 = C_5$ and $C_7 = LC_3/2$.

Rearranging to isolate the gradient term:

$$\begin{aligned}\frac{C_4}{2}\eta K\mathbb{E}[\|\nabla\mathcal{F}(\Theta_{g,t})\|^2] \leq \mathcal{F}(\Theta_{g,t}) - \mathbb{E}[\mathcal{F}(\Theta_{g,t+1})] \\ + C_6\eta K(\bar{\Delta}_t^2 + \zeta^2 + \sigma^2/K) + C_7\eta^2K^2(\sigma^2 + \bar{\Delta}_t^2 + \zeta^2)\end{aligned} \tag{57}$$

Summing this inequality over $t = 0, 1, \ldots, T-1$:

$$\begin{aligned}\frac{C_4}{2}\eta K\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{F}(\Theta_{g,t})\|^2] \leq \sum_{t=0}^{T-1}(\mathcal{F}(\Theta_{g,t}) - \mathbb{E}[\mathcal{F}(\Theta_{g,t+1})]) \\ + \sum_{t=0}^{T-1}\left[C_6\eta K(\bar{\Delta}_t^2 + \zeta^2 + \sigma^2/K) + C_7\eta^2K^2(\sigma^2 + \bar{\Delta}_t^2 + \zeta^2)\right]\end{aligned} \tag{58}$$

The first term on the right-hand side is a telescoping sum:

$$\sum_{t=0}^{T-1}(\mathcal{F}(\Theta_{g,t}) - \mathbb{E}[\mathcal{F}(\Theta_{g,t+1})]) = \mathcal{F}(\Theta_{g,0}) - \mathbb{E}[\mathcal{F}(\Theta_{g,T})] \tag{59}$$

Assuming the objective function is bounded below by F^*, i.e., $\mathcal{F}(\Theta) \geq F^*$ for all Θ, we have $\mathcal{F}(\Theta_{g,0}) - \mathbb{E}[\mathcal{F}(\Theta_{g,T})] \leq \mathcal{F}(\Theta_{g,0}) - F^*$.

Substituting this back and dividing by T:

$$\begin{aligned}\frac{C_4\eta K}{2T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{F}(\Theta_{g,t})\|^2] \leq \frac{\mathcal{F}(\Theta_{g,0}) - F^*}{T} \\ + \frac{1}{T}\sum_{t=0}^{T-1}\left[C_6\eta K(\bar{\Delta}_t^2 + \zeta^2 + \sigma^2/K) + C_7\eta^2K^2(\sigma^2 + \bar{\Delta}_t^2 + \zeta^2)\right]\end{aligned} \tag{60}$$

Finally, dividing by $\frac{C_4\eta K}{2}$ yields the bound on the average squared gradient norm:

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{F}(\Theta_{g,t})\|^2] \leq \frac{2(\mathcal{F}(\Theta_{g,0})-F^*)}{C_4\eta KT} \\ + \frac{2C_6}{C_4}\left(\frac{1}{T}\sum_{t=0}^{T-1}\bar{\Delta}_t^2+\zeta^2+\frac{\sigma^2}{K}\right) \\ + \frac{2C_7}{C_4}\eta K\left(\frac{1}{T}\sum_{t=0}^{T-1}(\sigma^2+\bar{\Delta}_t^2+\zeta^2)\right)\end{aligned} \tag{61}$$

Let $\bar{\Delta}^2 = \frac{1}{T}\sum_{t=0}^{T-1}\bar{\Delta}_t^2$ be the average model approximation error over T rounds. Define constants $C_8 = 2/C_4$, $C_9 = 2C_6/C_4$, and $C_{10} = 2C_7/C_4$. We arrive at the final convergence result:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{F}(\Theta_{g,t})\|^2] \leq \underbrace{\frac{C_8(\mathcal{F}(\Theta_{g,0})-F^*)}{\eta KT}}_{\text{Vanishing term}} + \underbrace{C_9\left(\bar{\Delta}^2+\zeta^2+\frac{\sigma^2}{K}\right)+C_{10}\eta K(\sigma^2+\bar{\Delta}^2+\zeta^2)}_{\text{Error floor}} \tag{62}$$

This result shows that the average squared gradient norm converges to an error floor determined by the data heterogeneity (ζ^2), stochastic gradient noise (σ^2/K), average model approximation error ($\bar{\Delta}^2$), and the chosen learning rate (η), at a rate of $\mathcal{O}(1/T)$. □

C Additional Experimental Results and Analyses

This appendix provides supplementary materials and detailed results that support the claims made in the main paper. The content is organized to present further validation of LEFF's scalability, a comprehensive analysis of client-side overhead, ablation studies on key hyperparameters, and empirical validation of our theoretical convergence claims.

C.1 Scalability to State-of-the-Art Models and Benchmarks

To demonstrate the scalability and effectiveness of LEFF on contemporary large-scale architectures, we extend our evaluation to the Llama-3.1-8B Grattafiori et al. (2024) model using the challenging MMLU benchmark Hendrycks et al. (2021). This experiment includes highly relevant state-of-the-art baselines FLoRA Wang et al. (2024d) and FlexLoRA Bai et al. (2024). As shown in Table 4, LEFF outperforms these recent methods by a significant margin of 1.8-2.3 points on the 5-shot MMLU average. This superior performance is achieved while simultaneously reducing the peak client-side GPU memory by over 36%. These results confirm that LEFF's design is highly effective and its efficiency benefits are even more pronounced on larger models where client-side resources are the primary bottleneck.

Table 4: Comparison of LEFF with SOTA methods on the Llama-3.1-8B model using the MMLU benchmark (5-shot average accuracy).

Method	MMLU (5-shot Avg.)	Peak GPU Memory (Client)
FlexLoRA	55.7	46.868 GB
FLoRA	55.2	46.868 GB
LEFF	57.5	29.881 GB

C.2 Analysis of Client-Side Computational Overhead

A detailed analysis of the computational overhead on client devices is crucial for evaluating the practical viability of any federated fine-tuning method. We present a comprehensive comparison of the number of trainable parameters and the peak GPU memory usage across various models and algorithms in Table 5. This analysis highlights LEFF's unique and superior efficiency profile. LEFF drastically reduces peak GPU memory—for instance, achieving a 79% reduction on GPT2-Large

compared to FedAvg and 66% compared to FedLoRA. This is because its architecture selectively loads only the necessary layers into GPU memory, whereas other PEFT methods must load the entire frozen model, resulting in much higher memory footprints. Concurrently, while achieving the lowest memory cost, LEFF deliberately retains more trainable parameters than methods like FedLoRA or FedBitFit. This design maintains greater model expressiveness, which translates to superior task performance. Thus, LEFF strikes an optimal balance, delivering the minimal resource cost for deployment on constrained devices while preserving enough capacity for high-accuracy fine-tuning.

Table 5: Computational cost comparison of different models and algorithms. "OOM" indicates Out of Memory. For Llama-3.1-8B, FedBitFit is not applicable as the model has no bias terms to train.

Model	Algorithm	Trainable Params	Peak Memory (GB)
DeBERTaV3-Base	FedAvg	85,648,130	3.841
DeBERTaV3-Base	FedLoRA	1,340,930	2.644
DeBERTaV3-Base	FedBitFit	102,914	2.198
DeBERTaV3-Base	LEFF	7,681,538	2.136
DeBERTaV3-Base	SLoRA	1,340,930	2.644
DeBERTaV3-Large	FedAvg	303,363,074	9.361
DeBERTaV3-Large	FedLoRA	3,557,378	6.660
DeBERTaV3-Large	FedBitFit	272,386	5.488
DeBERTaV3-Large	LEFF	13,649,922	3.005
DeBERTaV3-Large	SLoRA	3,557,378	6.660
GPT2	FedAvg	85,056,000	3.358
GPT2	FedLoRA	811,008	2.476
GPT2	FedBitFit	102,144	2.206
GPT2	LEFF	7,089,408	1.719
GPT2	SLoRA	811,008	2.476
GPT2-Large	FedAvg	708,390,400	15.548
GPT2-Large	FedLoRA	4,055,040	9.739
GPT2-Large	FedBitFit	508,160	8.318
GPT2-Large	LEFF	19,680,000	3.240
GPT2-Large	SLoRA	4,055,040	9.739
Llama-3.1-8B	FedAvg	OOM	OOM
Llama-3.1-8B	FedLoRA	20,971,520	46.868
Llama-3.1-8B	FedBitFit	No Bias	No Bias
Llama-3.1-8B	LEFF	743,452,672	29.881
Llama-3.1-8B	SLoRA	20,971,520	46.868

C.3 Ablation Study on the Proxy Dataset

The choice of the proxy dataset for knowledge distillation is a key aspect of the LEFF framework. We conduct an ablation study to investigate the sensitivity of our method to this choice. Our distillation objective is designed to be robust by matching intermediate functional representations (hidden states and attention matrices) rather than task-specific knowledge, which should reduce sensitivity to the proxy data's domain. The empirical results on the E2E NLG task, shown in Table 6, confirm this hypothesis. Performance remains remarkably stable across diverse corpora, from the in-domain WebNLG to general-purpose datasets like WikiText-103 and OpenWebText. The minimal fluctuation across all metrics demonstrates that LEFF is not reliant on a perfectly-matched proxy corpus, validating its practical applicability in real-world scenarios where such data may not be readily available.

Table 6: Ablation study on the choice of proxy datasets for the E2E NLG task.

Proxy Dataset	BLEU	NIST	METEOR	ROUGE	CIDEr
WikiText-103	0.5765	8.0012	0.4041	0.6310	1.7450
WebNLG	0.5799	8.0296	0.4064	0.6346	1.7521
OpenWebText	0.5712	7.9688	0.4015	0.6259	1.7345

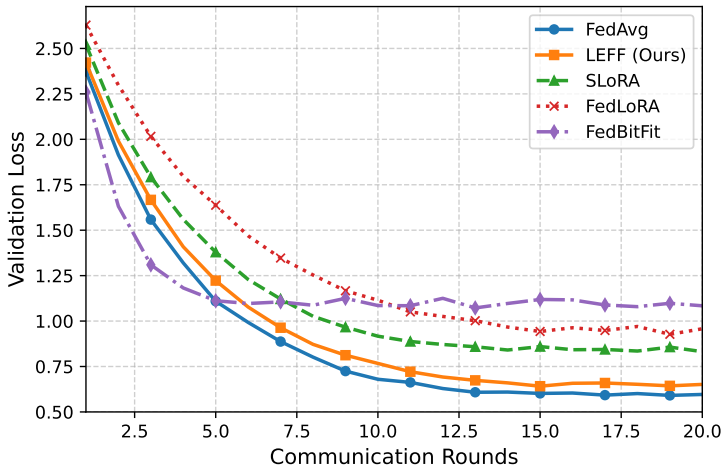

Figure 8: The validation loss convergence curve of different methods under high data heterogeneity ($\alpha = 0.05$).

C.4 Statistical Stability of Results

To ensure the reliability of our findings, we report the standard deviations of our main experimental results across three runs with different random seeds. The results, summarized in Table 7, show that the standard deviations are consistently small across all tasks and methods. This confirms that our method is stable and the reported performance gains are consistent. LEFF exhibits stability comparable to or better than the baselines, substantiating the robustness of our conclusions.

Table 7: Standard deviations of test results for DeBERTaV3 (on GLUE tasks) and GPT2 (on the E2E NLG task) across three runs with different random seeds.

	DeBERTaV3 (GLUE Tasks)								GPT2 (E2E NLG Task)				
Model	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	BLEU	NIST	METEOR	ROUGE	CIDEr
FedAvg	1.52	0.71	0.88	0.45	0.31	0.53	0.62	1.21	0.0041	0.1512	0.0028	0.0033	0.0805
FedBitFit	1.89	0.95	1.02	0.82	0.55	0.98	0.77	1.53	0.0053	0.2144	0.0045	0.0049	0.1231
FedLoRA	1.72	0.88	1.15	0.75	0.49	1.05	0.81	1.68	0.0058	0.2516	0.0041	0.0051	0.1189
SLoRA	1.45	0.81	0.95	0.68	0.42	0.85	0.71	1.35	0.0049	0.1832	0.0035	0.0039	0.0948
LEFF	1.38	0.75	0.91	0.51	0.35	0.62	0.68	1.25	0.0045	0.1604	0.0031	0.0035	0.0882

C.5 Empirical Convergence Analysis

Figure 8 provides an empirical validation of our theoretical convergence analysis presented in Section 4. The validation loss curves for all compared methods were plotted under a high data heterogeneity setting ($\alpha = 0.05$). The results show that LEFF converges stably and efficiently. Its final validation loss is substantially closer to that of full fine-tuning (FedAvg) compared to other PEFT methods like FedLoRA and SLoRA. This empirical evidence supports our theoretical analysis (Theorem 1), demonstrating that LEFF not only converges but also reaches a superior solution in challenging non-IID environments.

D Broader Impacts

D.1 Environmental Cost

The server-side computation in LEFF for layer importance calculation and knowledge distillation represents a trade-off for enabling client-side efficiency. By making fine-tuning feasible on numerous distributed, low-power edge devices, our approach can reduce the overall system's reliance on large, consistently energy-intensive data centers. This decentralization lowers the barrier to entry for

participation in model training and may lead to a more distributed and potentially more energy-efficient computational footprint compared to fully centralized training paradigms.

D.2 Dual-Use Risks

Making LLM fine-tuning more efficient and accessible could lower the barrier for misuse. However, the federated architecture of LEFF provides a natural control and governance point at the central server. This centralized aggregation step allows for the implementation of safeguards, such as client vetting, anomaly detection in model updates, and monitoring for malicious fine-tuning objectives. Such governance features are largely absent in purely decentralized or local fine-tuning scenarios, providing a mechanism to mitigate potential misuse.