EcoLoRA: Communication-Efficient Federated Fine-Tuning of Large Language Models

Anonymous ACL submission

Abstract

To address data locality and privacy restrictions, Federated Learning (FL) has recently been adopted to fine-tune large language models (LLMs), enabling improved performance on various downstream tasks without requiring aggregated data. However, the repeated exchange of model updates in FL can result in prohibitively high communication costs, hindering the distributed learning process. To address this challenge, we propose EcoLoRA, a novel communication-efficient federated fine-tuning framework for LLMs. Leveraging the modular structure, we propose a round-robin segment sharing scheme, where each client uploads only a complementary LoRA segment per round to reduce network bandwidth. It is further combined with adaptive sparsification methods tailored to LoRA's training dynamics and lossless encoding techniques. We conduct extensive evaluations on both question-answering and value-alignment tasks across multiple datasets and models. The results show that EcoLoRA significantly reduces communication overhead without compromising performance. For instance, it reduces communication time by up to 79% and total training time by up to 65%.

1 Introduction

With the advancements in scaling laws (Kaplan et al., 2020), the parameter sizes of pre-trained language models have grown exponentially (Chowdhery et al., 2023). Despite this rapid expansion, large language models (LLMs) remain constrained by their inherent knowledge boundaries, limiting their effectiveness in certain downstream tasks (Mao et al., 2024). These limitations necessitate task-specific fine-tuning. However, the substantial data required for fine-tuning is often distributed across multiple entities, raising significant privacy concerns related to data sharing.

Federated fine-tuning has emerged as a promising approach to mitigate these concerns. Recent studies have largely focused on integrating parameter-efficient fine-tuning (PEFT) methods into federated learning (FL) to reduce computational costs (Che et al., 2023; Cho et al., 2024; Babakniya et al.; Zhang et al., 2024; Sun et al., 2024; Bai et al., 2024), where a widely adopted strategy involves transmitting low-rank adaptation (LoRA) modules during the FL process. While LoRA significantly reduces the number of parameters exchanged compared to full fine-tuning, the massive scale of LLMs means that even these modules remain relatively large. Furthermore, repeatedly exchanging these modules during multiple communication rounds results in prohibitively high communication costs, making communication the essential bottleneck in training time.

Such prohibitive overhead can significantly hinder the participation of diverse clients, a key foundation for federated learning. More specifically, network connection speeds and their associated costs vary significantly across different areas, often differing by orders of magnitude (Howdle, 2023). For instance, many less-developed countries achieve bandwidths below 2 Mbps (Sumra, 2024), and rural areas often suffer from even poorer connections due to limited infrastructure. These disparities can prevent a large percentage of participants from contributing to FL due to expensive and unstable connectivity, excluding valuable high-quality data and undermining fairness in the learning process (Dorfman et al., 2023). Furthermore, network speeds are highly asymmetric, with upload speeds often being significantly slower than download speeds (Konečný, 2016), which presents additional challenges for FL.

In this work, we propose EcoLoRA, a novel **E**fficient **Co**mmunication framework specifically tailored to the unique training strategies and dynamics of federated fine-tuning of LLMs. First, leveraging the modular structure of LoRA, we introduce a round-robin segment-sharing scheme in

which each client transmits only a complementary portion of the LoRA module rather than the entire module. Second, we propose an adaptive sparsification technique customized for the distinct training dynamics observed in matrices A and B of LoRA. Third, the adaptive sparsification method naturally enables parameter distribution suitable for geometric compression, allowing us to employ Golomb coding for further communication efficiency.

To demonstrate the effectiveness of EcoLoRA, we incorporate it into various state-of-the-art methods across different tasks (including general question answering and value alignment), datasets, and models. Our results show significant communication savings while preserving model performance. Notably, EcoLoRA reduces uploaded parameters by up to 89% and overall communication parameters by up to 58% compared to existing approaches. Under practical network conditions, it reduces communication time by up to 79% and total training time by 65%. Moreover, our approach remains robust under various non-i.i.d. data conditions and adds only minimal computational overhead.

Our contributions are summarized as follows:

- We propose a novel framework, EcoLoRA, a communication-efficient federated fine-tuning framework for LLMs.
- We provide a theoretical proof of the convergence of EcoLoRA.
- We conduct extensive experiments, demonstrating that EcoLoRA significantly reduces communication overhead while preserving accuracy.

2 Related Work

2.1 Parameter-efficient Fine-tuning of LLMs

The large parameter size of LLMs often makes traditional full fine-tuning prohibitively expensive. To address this challenge, various parameter-efficient fine-tuning (PEFT) techniques have been proposed, including prefix-tuning (Li and Liang, 2021), prompt-tuning (Lester et al., 2021), and adapter-based methods (Hu et al., 2023). Among these approaches, low-rank adaptation (LoRA) (Hu et al., 2022), which leverages low-rank matrices to re-parameterize pre-trained weight matrices, has received unprecedented attention. LoRA requires tuning less than 1% of the parameters needed for a full fine-tune while still achieving comparable performance across a wide range of downstream tasks, without introducing additional inference latency. Building on these advantages, numerous LoRA variants have been developed to further improve its efficiency and accuracy (Kopiczko et al., 2023; Zhang et al., 2023; Liu et al., 2024).

2.2 Federated Fine-tuning of LLMs

Federated fine-tuning of LLMs has recently received significant attention. Most of the existing work has focused on integrating PEFT methods into federated learning to reduce computation costs (Che et al., 2023; Wu et al., 2024; Cho et al., 2024; Babakniya et al.; Zhang et al., 2024; Sun et al., 2024; Bai et al., 2024). For example, Zhang et al. (2024) incorporate LoRA into the FedAvg framework so that only LoRA modules need to be trained and aggregated. Extending this approach to resource-constrained and heterogeneous scenarios, Wang et al. (2024) propose a stacking-based aggregation strategy for heterogeneous LoRA modules, where individual LoRA modules are uploaded for aggregation, and the resulting stacked full-size LoRA weights are distributed back to clients. Sun et al. (2024) further enhance performance under differential privacy guarantees and improve computational efficiency by fine-tuning only the zero-initialized LoRA matrices. Although these approaches reduce both computation and communication costs compared to full fine-tuning, transmitting LoRA modules still imposes considerable overhead. Even though LoRA accounts for a small portion of the total parameters, the massive scale of LLMs means these modules remain large. Repeatedly exchanging them during multiple communication rounds results in prohibitively high communication costs, making communication the dominant bottleneck in training time.

Another line of research leverages zeroth-order optimization methods for federated LLM fine-tuning (Qin et al., 2024; Xu et al., 2024). While these approaches improve communication efficiency, their reliance on zeroth-order optimization significantly reduces computational efficiency compared to backpropagation-based methods. Consequently, these techniques substantially increase the computation time and, therefore, prolong the overall training process, particularly in scenarios with limited participating clients or resource-constrained environments.

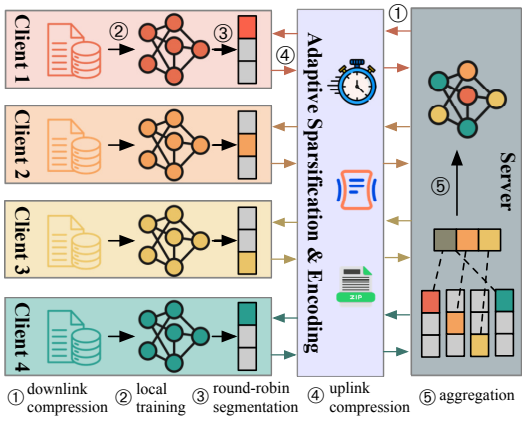


Figure 1: Overview of our proposed EcoLoRA.

2.3 Communication Optimization in FL

Communication optimization in traditional federated learning has drawn considerable attention, primarily through three techniques: quantization, sparsification, and client sampling. Quantization methods compress model parameters by representing them with fewer bits (Bernstein et al., 2018; Leng et al., 2018; Xu et al., 2020; Horvóth et al., 2022). However, quantization typically offers limited compression and may lead to noticeable accuracy degradation, particularly in non-IID settings. Sparsification methods generally achieve higher compression ratios by transmitting sparse representations of model parameters (Aji and Heafield, 2017; Tsuzuku et al., 2018; Sahu et al., 2021). A representative sparsification technique, top-k sparsification (Aji and Heafield, 2017), selects parameters based on magnitude and has demonstrated robustness to non-IID data distributions. Lastly, client sampling approaches selectively include clients based on their expected contributions to model improvement by employing carefully designed criteria (Luping et al., 2019; Sun et al., 2019; Tang et al., 2022).

3 Method

3.1 Problem Formulation

We consider an FL setting with one server and K devices. Each device i holds a local dataset $\mathcal{D}_i = (x_j, y_j)^{n_i}$, where n_i, x_j, y_j denote the number of samples, the input samples, and labels in client i, respectively. The total number of samples across all devices is $N = \sum_{i=1}^{K} n_i$. Following recent state-of-the-art approaches, the pre-trained LLMs $\mathcal{M}$ remain fixed on each device, while only the LoRA parameters are updated and exchanged between device i and the server. The optimization goal is to find a set of LoRA parameters $\mathcal{P}$ to minimize the global loss:

$$\min_{\mathcal{P}} \quad F(\mathcal{M}, \mathcal{P}, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{K} n_i \, \mathbb{E}_{(x_j, y_j) \sim \mathcal{D}_i} \left[\mathcal{L}(\mathcal{M}, \mathcal{P}, x_j, y_j) \right], \tag{1}$$

where $\mathcal{L}(\mathcal{M}, \mathcal{P}, x_j, y_j)$ is the loss evaluated by the model $\mathcal{M}$ with LoRA parameters $\mathcal{P}$ on the local data (x_j, y_j). LoRA models the weight update $\Delta W \in \mathbb{R}^{m \times n}$ through a low-rank decomposition BA, where $B \in \mathbb{R}^{m \times r}$ and $A \in \mathbb{R}^{r \times n}$ are two low-rank matrices with $r \ll \min(m, n)$.

3.2 System Model

Our primary objective is to enhance communication efficiency specifically for federated fine-tuning of LLMs. This setting, however, differs significantly from traditional FL paradigms, particularly due to the use of parameter-efficient fine-tuning techniques, which result in distinct training dynamics and parameter distributions. As a result, conventional communication optimization methods, such as top-k sparsification, may fail to fully leverage these unique properties, leading to suboptimal communication gains. Other methods, like active client sampling (Tang et al., 2022), may introduce considerable computational overhead, which undermines their practicality in LLM fine-tuning scenarios. These limitations highlight the need for communication-efficient techniques specifically tailored to the characteristics of federated LLM fine-tuning. To guide the design of such methods, we establish the following system goals:

- *Communication Efficiency:* The framework should substantially reduce communication overhead while preserving model performance.
- *Minimal Computational Overhead:* Since LLM fine-tuning already incurs high computational costs, particularly on resource-constrained edge devices, our framework should introduce minimal additional overhead.
- *Robustness to Non-i.i.d. Data:* Because data distributions can vary significantly across clients in real-world settings, our framework should remain robust under non-i.i.d. conditions.

To address these challenges, we propose EcoLoRA, a novel communication-efficient FL framework illustrated in Figure 1. First, we propose a round-robin segment-sharing scheme, leveraging the modular structure of LoRA. Instead of transmitting the entire LoRA module, each client shares

only a complementary portion, significantly reducing communication overhead. Second, we introduce an adaptive sparsification method tailored for the different training dynamics observed in matrices A and B. This method dynamically compresses parameters based on their specific training behavior, ensuring minimal performance degradation. Third, the adaptive sparsification method naturally enables a parameter distribution suitable for geometric compression, which we exploit through Golomb coding to further optimize communication efficiency. We elaborate round-robin segment sharing, adaptive sparsification, and encoding in Sections 3.3, 3.4, and 3.5, respectively. Additionally, we analyze computational overhead in Section 3.6 and provide convergence analysis in Section 3.7.

3.3 Round-Robin Segment Sharing

LoRA can be treated as a modular plug-in to the base model as each LoRA module can be independently attached or removed. Leveraging this modularity, we propose a novel round-robin segment sharing scheme to reduce communication costs, where each client only shares a portion of its LoRA parameters in each round. Formally, we partition the LoRA parameters across all layers into N_s equally sized segments, denoted as $\mathcal{P} = [s_0, s_1, \ldots, s_{N_s-1}]$. In each training round t, each client i uploads only one segment, with the ID identified by $(i + t) \bmod N_s$. To ensure that all segments are uploaded by at least one client in each round, enabling complete LoRA parameter updates, we further require $N_s \leq N_t$, where N_t is the number of participating clients per round.

At the server side, segments with the same ID are aggregated by a weighted average, and the global LoRA model is reassembled from these aggregated segments. Let $\mathcal{P}^t$ denote the aggregated global LoRA model in the t-th round, $s^t_{i,s}$ represent the s-th segment uploaded by the i-th client in the t-th round, c^k denote the set of clients who upload the k-th segment, and n_i represent the number of samples in client i. The aggregation rule is:

$$\mathcal{P}^t = \left[\frac{\sum_{i \in c^0} n_i s^t_{i,0}}{\sum_{i \in c^0} n_i}, \frac{\sum_{i \in c^1} n_i s^t_{i,1}}{\sum_{i \in c^1} n_i}, \ldots, \frac{\sum_{i \in c^{N_s-1}} n_i s^t_{i,N_s-1}}{\sum_{i \in c^{N_s-1}} n_i} \right], \tag{2}$$

For example, consider $N_t = 5$ clients and $N_s = 3$ segments. In round $t = 0$, client 0 uploads the segment with ID $(0 + 0) \bmod 3 = 0$, i.e., $s^0_{0,0}$; client 1 uploads $s^0_{1,1}$; client 2 uploads $s^0_{2,2}$; client 3 uploads $s^0_{3,0}$; and client 4 uploads $s^0_{4,1}$. The server then averages $s^0_{0,0}$ and $s^0_{3,0}$ to form the 0-th segment, averages $s^0_{1,1}$ and $s^0_{4,1}$ to form the 1-th segment, and takes $s^0_{2,2}$ for the 2-th segment. Because each client transmits only a single segment in each round, this round-robin segment sharing scheme reduces the upload communication load to $1/N_s$ of the total parameters.

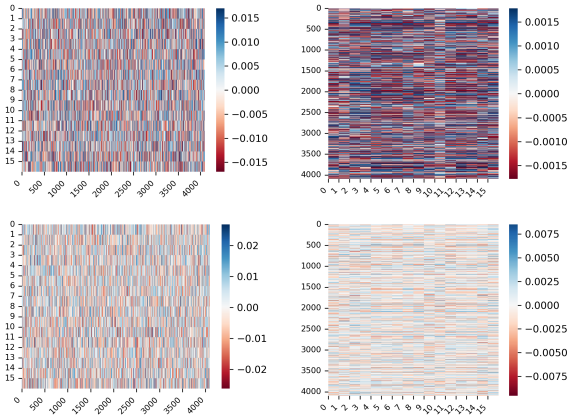
However, this partial update approach introduces a delay for segments that are not uploaded in a given round, which can increase the number of rounds required to converge. To mitigate potential accuracy degradation, we leverage the local model by taking a weighted average of the global and local models at the beginning of each round before optimization. This ensures that even if a segment is not uploaded in a particular round, its previous state still guides local optimization. Moreover, by mixing the globally shared model (the consensus among clients) with the client's locally fine-tuned model (adapted to its specific data), we improve robustness under non-i.i.d. distributions. In cross-device settings, only a subset of clients participates in each round, which may result in some clients remaining idle for many rounds and thus suffering from stale local parameters that potentially hamper global convergence (Xie et al., 2019) when using the simple average. To address this, we employ an exponential decay weighting (Chen et al., 2019) to update the local LoRA model:

$$\hat{\mathcal{P}}^t_i = (1 - e^{-\beta(t-\tau)})\mathcal{P}^t + e^{-\beta(t-\tau)}\mathcal{P}^\tau_i, \tag{3}$$

where t denotes the current global round, τ is the most recent round in which client i participated, and β is a hyperparameter balancing staleness.

3.4 Adaptive Sparsification

To further reduce the communication load for both uploading and downloading, we can adopt the sparsification techniques that have been successfully applied in traditional FL (Aji and Heafield, 2017). These techniques exploit the observation that most gradient updates are near zero. Among various sparsification approaches, top-k sparsification has demonstrated promising performance with non-i.i.d. data (Sattler et al., 2019) by selecting parameters with the highest k portion of magnitudes for transmission. Since the LoRA module in fine-tuning acts as parameter updates for LLMs, we analyzed matrices A and B during FL training to validate whether the LoRA updates also exhibit similar sparsity in federated LLM fine-tuning, following



(a) Matrix A (b) Matrix B

Figure 2: Visualization of LoRA matrices A and B at epochs 1 (top) and 20 (bottom) during FL training.

the experimental setup in Section 4.1. Figure 2 shows an example at epoch 0 and epoch 20. Two notable trends emerge from this analysis: (1) As training progresses, both LoRA matrices become sparser, with the many remaining values growing larger in magnitude. (2) Matrices A and B evolve differently; in particular, B becomes much sparser than A. To quantify this, we calculated the Gini coefficient, a statistical measure of distribution inequality where larger values indicate a higher proportion of extreme values. In epoch 0, matrix A had a coefficient of 0.337 and matrix B had 0.243, while by epoch 20, these values reached 0.359 and 0.406 respectively. These characteristics present unique opportunities for sparsification. First, to adapt to increasing sparsity, we propose time-adaptive top-k sparsification. We use the loss signal to scale k with training progress, as it both indicates training status and requires no additional computation:

$$k^t = k_{\min} + (k_{\max} - k_{\min}) \cdot e^{-\gamma(L_0 - L_{t-1})}, \quad (4)$$

where k^t is the sparsity level for round t, L_0 is the initial loss, L_{t-1} is the global loss for round $t-1$, and $k_{\max}$ and $k_{\min}$ define the sparsification range. As training loss decreases, k^t is reduced, reflecting that the model has learned sufficient knowledge and updates have become sparser. Second, to address the distinct patterns in matrices A and B, we introduce a matrix-adaptive sparsification scheme. We set smaller $k_{\min}$ value for B (due to its higher sparsity) and use a larger γ for B to capture its rapid change in sparsity.

To mitigate information loss during sparsification, we locally accumulate untransmitted updates as residuals until they become large enough for transmission. Let SC_k denote top-k sparsification, the compressed parameter $\hat{\mathcal{P}}^{t+1}$ is computed as:

$$\hat{\mathcal{P}}^{t+1} = \mathrm{SC}_{k^{t+1}}(\mathcal{P}^{t+1} + R^t), \quad (5)$$

where R^t is the residue at round t. We then update the residue as:

$$R^{t+1} = R^t + \mathcal{P}^{t+1} - \hat{\mathcal{P}}^{t+1}. \quad (6)$$

R is initialized as an empty residual at the beginning of the training. This approach ensures that large updates are transmitted immediately while eventually sending all updates over time.

3.5 Lossless Encoding

To communicate the set of sparse LoRA tensors between the server and the client, we only need to transmit the positions of the nonzero elements in the flattened tensors, along with a one-bit sign and 16-bit values (assuming FP16) for each nonzero update. However, the positions can still be expensive to communicate because they are typically stored with a fixed number of 16 bits. From an information-theoretic perspective, we can further compress these positions using lossless encoding (Sattler et al., 2019). Specifically, rather than sending the absolute positions of the nonzero elements, we send the distances between consecutive nonzero positions. Given our adaptive sparsification rate k, each element is nonzero with probability k, thus the distance between two consecutive nonzero elements follows a geometric distribution with parameter k, where the probability of a distance of length n is $(1-k)^{n-1}k$. For such geometrically distributed nonnegative integers, Golomb coding provides an optimal encoding scheme by representing each number as a combination of quotient and remainder (Golomb, 1966). For example, when $k = 0.1$, using Golomb coding can reduce the average number of bits required to encode each nonzero position to $b^* = 4.8$, which leads to approximately a $3.3\times$ compression factor per position.

3.6 Analysis of Computational Overhead

We now analyze the additional computational overhead introduced by our proposed method. For round-robin segment sharing, we compute a weighted average of the global and local models in Eq. 3. Since this is an element-wise operation, it requires roughly $2|\mathcal{P}|$ operations. For adaptive sparsification, we could select the top-k LoRA updates using efficient selection algorithms, such as

Quicksort, which take about $O\big(|\mathcal{P}|\log(|\mathcal{P}|)\big)$. Additionally, untransmitted gradients are accumulated as residuals via simple element-wise additions, contributing $O\big(|\mathcal{P}|\big)$ cost. For lossless encoding, we first compute the differences between consecutive indices, which takes $O\big(k|\mathcal{P}|\big)$ time. We then apply Golomb coding to each gap, also running in linear time with respect to $k|\mathcal{P}|$. Overall, the per-round overhead scales nearly linearly with the number of LoRA parameters $|\mathcal{P}|$. Since $|\mathcal{P}|$ is typically much smaller than the full model size $|\mathcal{M}|$, the additional overhead remains minimal compared to the cost of forward and backward propagation.

3.7 Convergence Analysis

We now present the convergence analysis for EcoLoRA, adhering to the standard procedures described in Li et al. (2019). Our analysis relies on the following assumptions:

Assumption 1 (Smoothness). The objective function F is L-smooth, meaning:

$$F(P_{t+1}) \leq F(P_t) + \langle \nabla F(P_t), P_{t+1} - P_t \rangle + \frac{L}{2}\|P_{t+1} - P_t\|^2.$$

Assumption 2 (Bounded Gradients). The expected squared norm of the stochastic gradients is uniformly bounded by a constant G^2:

$$\mathbb{E}\big\|\nabla F(P_t)\big\|^2 \leq G^2.$$

Assumption 3 (Contractive Property). There exists a constant $\delta \in (0, 1]$ such that, for any x:

$$\|C(x) - x\|^2 \leq (1 - \delta)\|x\|^2.$$

We define the following constants:

$$\mu = \eta\left(\frac{5}{2} + \delta(2\eta L - 1) - 3\eta L\right),$$
$$\Delta = \frac{e^{-\beta}}{1 - e^{-\beta}} L^2 \eta^2 N_s^2 G^2. \tag{7}$$

Under these assumptions, selecting the learning rate within the interval $\frac{1}{L} < \eta < \frac{5-2\delta}{(6-4\delta)L}$, after T communication rounds, EcoLoRA satisfies:

$$\frac{1}{T}\sum_{t=0}^{T-1}\big\|\nabla F(P_t)\big\|^2 \leq \frac{F(P_0) - F^\star}{\mu T} + \frac{\eta(2\eta L - 1)\Delta}{\mu}.$$

Choosing $\eta = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$, we obtain the final convergence rate:

$$\frac{1}{T}\sum_{t=0}^{T-1}\big\|\nabla F(P_t)\big\|^2 = \mathcal{O}\left(T^{-1/2}\right)$$

The detailed proof is given in Appendix B.

4 Experiment

4.1 Experimental Setup

Models and Datasets. We consider two tasks: question answering (QA) and value alignment (VA). For QA, we use Llama2 (Touvron et al., 2023) with 7B and 13B parameters. For VA, we use the uncensored version of Vicuna-7B (Xu et al., 2023). As instruction datasets for QA, we adopt Databricks-dolly-15k (Conover et al., 2023) and Alpaca-GPT4 (Peng et al., 2023). For VA, we use the UltraFeedback dataset (Cui et al., 2024).

Evaluation Metrics. We measure both model accuracy and communication efficiency. For QA performance, we report results on the ARC easy and challenge benchmark (Clark et al., 2018), taking the average of both sets as the ARC score; for the VA task, we evaluate using MT-bench (Zheng et al., 2023) and MMLU (Hendrycks et al., 2020) following (Wang et al., 2024; Ye et al., 2024). We report communication parameters and time under simulated practical network conditions to assess communication efficiency.

Baselines. Our work proposes a general communication efficient framework to enhance existing federated LLM fine-tuning methods. To evaluate its effectiveness, we apply our framework to state-of-the-art approaches: FedIT (Zhang et al., 2024), FLoRA (Wang et al., 2024), and FFA-LoRA (Sun et al., 2024), and compare the resulting performance to the original methods.

FL Settings and Implementation Details. Following Zhang et al. (2024), we implement our framework in a federated learning environment with 100 clients. In each round, we randomly sample 10 clients and conduct training for 40 global rounds. To simulate realistic scenarios, we adopt a non-i.i.d. data distribution across clients. Detailed experimental configurations and hyperparameter settings are provided in Appendix A[1].

4.2 Evaluation Results

Results of QA Tasks. Table 1 shows the model accuracy on the ARC benchmark and the communication overhead for various methods, both with and without EcoLoRA. Our approach achieves performance comparable to the baseline while significantly reducing communication costs. For example, when applying our method to FFA-LoRA

[1]We will release our code upon publication of the paper.

Table 1: Comparison of accuracy and associated communication parameters (in millions) across different methods.

Model	Method	Alpaca			Dolly		
		ARC	Upload Param.	Total Param.	ARC	Upload Param.	Total Param.
Llama2-7B	FedIT	66.6	2520.1	5040.1	66.5	2772.1	5544.2
	FedIT w/ EcoLoRA	66.6	**346.5**	**2675.7**	66.5	**481.1**	**3765.6**
	FLoRA	67.0	2856.1	31416.9	66.4	2688.1	29568.8
	FLoRA w/ EcoLoRA	67.2	**350.9**	**24165.7**	66.3	**321.6**	**22023.9**
	FFA-LoRA	67.4	1512.0	3024.1	66.7	1260.0	2520.1
	FFA-LoRA w/ EcoLoRA	67.4	**160.1**	**1265.2**	66.7	**173.9**	**1346.1**
Llama2-13B	FedIT	70.3	3674.1	7348.2	70.1	2361.9	4723.8
	FedIT w/ EcoLoRA	70.4	**488.9**	**3775.4**	70.0	**427.4**	**3254.8**
	FLoRA	70.3	4461.4	49075.3	69.8	4067.7	44745.1
	FLoRA w/ EcoLoRA	70.5	**576.3**	**39816.7**	70.1	**555.8**	**38026.2**
	FFA-LoRA	70.2	2099.5	4199.0	69.9	2558.7	5117.5
	FFA-LoRA w/ EcoLoRA	70.2	**272.0**	**2137.5**	69.9	**261.5**	**1943.3**

(Sun et al., 2024) on Llama2-7B trained with Alpaca, we reduce the required upload communication by 89%. This reduction is particularly advantageous given that upload speeds are often far slower than download speeds (Konečný, 2016). Moreover, the total communication parameters are reduced by 58% under the same setting. Furthermore, the EcoLoRA framework has demonstrated generalizability across different methods, thereby expanding its applicability. For example, it can be combined with approaches that leverage heterogeneous client resources (Wang et al., 2024) or that strengthen performance under differential privacy constraints (Sun et al., 2024), allowing practitioners to benefit from the respective advantages of each approach.

Results of VA Tasks. Alignment with human preferences is a crucial step in LLM post-training (Lee et al., 2023). To evaluate EcoLoRA on this task, we implemented federated direct preference optimization (DPO) (Rafailov et al., 2023) following the approach of (Ye et al., 2024). Specifically, we use UltraFeedback as our local preference dataset; the response with the highest score is treated as the preferred response, and one of the remaining responses is randomly designated as the dispreferred response, following (Tunstall et al., 2023). As shown in Table 2, EcoLoRA substantially reduces both the upload and total communication parameters while achieving slightly higher performance on MT-bench and MMLU.

Table 2: Comparison of model accuracy and communication parameters (in millions) of federated DPO with and without EcoLoRA.

Method	MT-bench	MMLU	Upload P.	Total P.
DPO	3.26	34.8	1719.7	3439.3
w/ EcoLoRA	3.28	35.4	**348.8**	**2072.1**

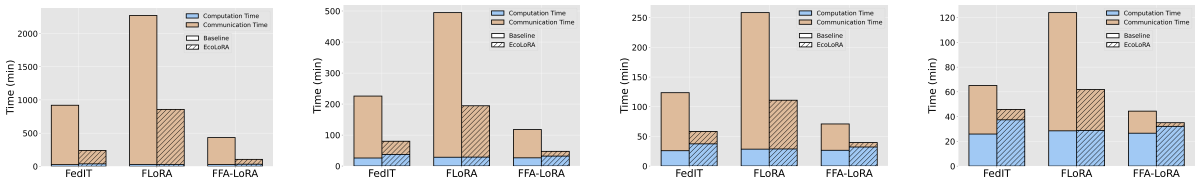
4.3 Evaluation in Practical Networks

To evaluate the performance of EcoLoRA under realistic network conditions, we implemented a simulated federated learning platform following (Ekaireb et al., 2022), using `ns-3`, a widely adopted discrete-event simulator for network communications (Henderson et al., 2008).

Following practical uplink (UL) and downlink (DL) bandwidth settings in (Konečný, 2016), we simulate four bandwidth scenarios: 0.2/1 Mbps, 1/5 Mbps, 2/10 Mbps, and 5/25 Mbps, with a fixed latency of 50ms to capture different network conditions. Figure 3 compares the computation and communication time of EcoLoRA against baselines under these scenarios, using Llama2-7B trained on Dolly. Our results demonstrate that as network conditions deteriorate, communication time increasingly dominates the total training time. This effect is particularly notable given that actual throughput typically falls short of theoretical bandwidth. These findings underscore the importance of developing communication-efficient fine-tuning methods. Across all conditions, EcoLoRA significantly reduces communication overhead while introducing minimal computational cost. For instance, under the 1/5 Mbps setting, it reduces communication time by 79% and total training time by 65%. Moreover, the additional per-round computation cost remains *below 3s*, making EcoLoRA a practical solution for resource-constrained environments.

4.4 Ablation Study

In this section, we analyze the impact of various design components and hyperparameter choices. We



(a) UL/DL: 0.2/1 Mbps (b) UL/DL: 1/5 Mbps (c) UL/DL: 2/10 Mbps (d) UL/DL: 5/25 Mbps

Figure 3: The computation and communication time of applying ECOLORA under different network conditions.

also provide a comprehensive comparison with top-k sparsification and present additional experiments under non-i.i.d. conditions in Appendix C.

Impacts of Design Components. We conducted an ablation study to investigate how each design component influences both model performance and communication time (both upload and total communication) using Llama2-7B trained on the Dolly dataset with FedIT w/ ECOLORA method. Specifically, we examine the following variants: (1) w/o Round-Robin (R.R.) Segment: The entire LoRA module is transmitted. (2) w/o Sparsification: The adaptive sparsification method is removed. (3) w/ Fixed Sparsification: A fixed sparsification ratio is used while keeping the overall communication cost identical to that in adaptive sparsification. (4) w/o Encoding: The lossless encoding scheme is excluded. Table 3 reports the final accuracy and communication time required to reach the target accuracy of 66.5 for each variant. As shown, each design component notably reduces both the uploading time and total communication time. Additionally using a fixed sparsification ratio results in a significant accuracy drop. This decline occurs because update patterns vary across different training stages and between matrices A and B, which exhibit different levels of robustness to sparsification.

Table 3: Accuracy and communication time for achieving the target accuracy (66.5 on ARC) under different ablations. ("–" indicates target not achieved.)

Method	ARC	Upload Time	Total Time
w/o R.R. Segment	66.5	72.6	106.2
w/o Sparsification	66.6	25.6	55.6
w/ Fixed Sparsification	66.1	-	-
w/o Encoding	66.5	29.5	68.9
Full	66.5	**18.2**	**42.6**

Impacts of Compression Levels. We examine how different compression levels influence model accuracy and communication overhead. In particular, we vary the number of segments N_s in the Round-Robin scheme, as well as the minimum top-k thresholds for matrices A and B ($k^A_{\min}$, $k^B_{\min}$), using Llama2-7B trained on the Dolly dataset with FedIT w/ ECOLORA method. Table 4 reports both the accuracy and the communication parameters required to reach a target accuracy under different compression levels. We observe that choosing a smaller N_s can improve model accuracy and thus reduce download communication overhead (because fewer rounds are needed to achieve the target accuracy). However, it also increases upload communication overhead. Conversely, setting N_s too large can degrade model accuracy. On the other hand, applying higher sparsity to matrix B than to matrix A (for example, $k^A_{\min} = 0.6$ and $k^B_{\min} = 0.25$) does not negatively affect model accuracy. As discussed in Section 3.4, the B matrix is intrinsically sparser than the A matrix. Practitioners should select compression levels achieving an optimal balance between communication costs and accuracy based on the specific network constraints.

Table 4: Accuracy and communication parameters for achieving the target accuracy (66.5 on ARC) under different compressions. ("–" indicates target not achieved.)

Method	ARC	Upload P.	Total P.
$\{N_s = 3, k^A_{\min} = 0.6, k^B_{\min} = 0.5\}$	66.6	688.9	3495.7
$\{N_s = 5, k^A_{\min} = 0.6, k^B_{\min} = 0.5\}$	66.5	481.1	3765.6
$\{N_s = 10, k^A_{\min} = 0.6, k^B_{\min} = 0.5\}$	66.0	-	-
$\{N_s = 5, k^A_{\min} = 0.6, k^B_{\min} = 0.25\}$	66.5	271.2	2464.7
$\{N_s = 5, k^A_{\min} = 0.3, k^B_{\min} = 0.5\}$	66.2	-	-

5 Conclusion

In this paper, we introduced ECOLORA, a novel communication-efficient federated fine-tuning framework for LLMs. Our approach comprises a round-robin segment sharing scheme, an adaptive sparsification method, and lossless encoding. Extensive evaluations on QA and VA tasks across diverse datasets and models show that ECOLORA substantially reduces communication overhead while maintaining accuracy. Moreover, it remains robust under non-i.i.d. settings and incurs minimal computational overhead.

Limitations

Due to constraints in computational resources and time, we focused on a single, representative federated learning setting, following standard practice in existing studies. In future work, we plan to explore multiple federated learning scenarios involving diverse client participations to further validate the generalizability of our approach. Additionally, we restricted our experiments to Llama models for question-answering tasks and the Vicuna model for value alignment tasks; investigating a wider range of models will be crucial for fully demonstrating the robustness and versatility of our method.

Ethical Considerations

We propose a communication-efficient federated learning framework designed to improve system efficiency while preserving data privacy. Additionally, all our experiments use public datasets, we have not identified any specific risks arising from this study. However, we remain mindful of potential privacy and security implications that may be associated with federated learning in general.

References

Alham Aji and Kenneth Heafield. 2017. Sparse communication for distributed gradient descent. In *EMNLP 2017: Conference on Empirical Methods in Natural Language Processing*, pages 440–445. Association for Computational Linguistics (ACL).

Sara Babakniya, Ahmed Roushdy Elkordy, Yahya H Ezzeldin, Qingfeng Liu, Kee-Bong Song, MOSTAFA EL-Khamy, and Salman Avestimehr. Slora: Federated parameter efficient fine-tuning of language models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*.

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. 2024. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. 2018. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR.

Tianshi Che, Ji Liu, Yang Zhou, Jiaxiang Ren, Jiwen Zhou, Victor Sheng, Huaiyu Dai, and Dejing Dou. 2023. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7871–7888.

Yang Chen, Xiaoyan Sun, and Yaochu Jin. 2019. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE transactions on neural networks and learning systems*, 31(10):4229–4238.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. 2024. Heterogeneous lora for federated fine-tuning of on-device foundation models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12903–12913.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. In *Forty-first International Conference on Machine Learning*.

Ron Dorfman, Shay Vargaftik, Yaniv Ben-Itzhak, and Kfir Yehuda Levy. 2023. Docofl: Downlink compression for cross-device federated learning. In *International Conference on Machine Learning*, pages 8356–8388. PMLR.

Emily Ekaireb, Xiaofan Yu, Kazim Ergun, Quanling Zhao, Kai Lee, Muhammad Huzaifa, and Tajana Rosing. 2022. ns3-fl: Simulating federated learning with ns-3. In *Proceedings of the 2022 Workshop on ns-3*, pages 97–104. Association for Computing Machinery.

Solomon Golomb. 1966. Run-length encodings (corresp.). *IEEE transactions on information theory*, 12(3):399–401.

Thomas R Henderson, Mathieu Lacage, George F Riley, Craig Dowell, and Joseph Kopena. 2008. Network simulations with the ns-3 simulator. *SIGCOMM demonstration*, 14(14):527.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Samuel Horvóth, Chen-Yu Ho, Ludovit Horvath, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. 2022. Natural compression for distributed deep learning. In *Mathematical and Scientific Machine Learning*, pages 129–141. PMLR.

Dan Howdle. 2023. The cost of 1gb of mobile data in 237 countries.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Jakub Konečnỳ. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Dawid J Kopiczko, Tijmen Blankevoort, and Yuki M Asano. 2023. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback.

Cong Leng, Zesheng Dou, Hao Li, Shenghuo Zhu, and Rong Jin. 2018. Extremely low bit neural network: Squeeze the last bit out with admm. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

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

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Shih-yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

WANG Luping, WANG Wei, and LI Bo. 2019. Cmfl: Mitigating communication overhead for federated learning. In *2019 IEEE 39th international conference on distributed computing systems (ICDCS)*, pages 954–964. IEEE.

Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. 2024. A survey on lora of large language models. *arXiv preprint arXiv:2407.11046*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Zhen Qin, Daoyuan Chen, Bingchen Qian, Bolin Ding, Yaliang Li, and Shuiguang Deng. 2024. Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes. In *Forty-first International Conference on Machine Learning*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Atal Sahu, Aritra Dutta, Ahmed M Abdelmoniem, Trambak Banerjee, Marco Canini, and Panos Kalnis. 2021. Rethinking gradient sparsification as total error minimization. *Advances in Neural Information Processing Systems*, 34:8133–8146.

Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413.

Husain Sumra. 2024. Best and worst countries for wi-fi access.

Jun Sun, Tianyi Chen, Georgios Giannakis, and Zaiyue Yang. 2019. Communication-efficient distributed learning via lazily aggregated quantized gradients. *Advances in Neural Information Processing Systems*, 32.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. 2024. Improving lora in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*.

Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen. 2022. Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10102–10111.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yusuke Tsuzuku, Hiroto Imachi, and Takuya Akiba. 2018. Variance-based gradient compression for efficient distributed deep learning. *arXiv preprint arXiv:1802.06058*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. 2024. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Feijie Wu, Zitao Li, Yaliang Li, Bolin Ding, and Jing Gao. 2024. Fedbiot: Llm local fine-tuning in federated learning without full model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3345–3355.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. 2019. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Jinjin Xu, Wenli Du, Yaochu Jin, Wangli He, and Ran Cheng. 2020. Ternary compression for communication-efficient federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(3):1162–1176.

Mengwei Xu, Dongqi Cai, Yaozong Wu, Xiang Li, and Shangguang Wang. 2024. {FwdLLM}: Efficient federated finetuning of large language models with perturbed inferences. In *2024 USENIX Annual Technical Conference (USENIX ATC 24)*, pages 579–596.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. 2024. Openfedllm: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6137–6147.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. 2024. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6915–6919. IEEE.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Experimental Settings

The Alpaca-GPT4 dataset contains 52K instruction-following examples generated by GPT-4 using Alpaca prompts. The Dolly dataset consists of 15K text samples created by Databricks employees. The UltraFeedback dataset comprises 64K instructions.

To simulate non-i.i.d. data distribution across clients, we divide the datasets using a Dirichlet distribution with $\alpha = 0.5$. For the Dolly dataset, we directly use the provided category labels for splitting. Since the Alpaca dataset lacks explicit categories, we generate synthetic ones. Specifically, we concatenate the 'instruction' and 'input' fields of each sample into a single string, convert these strings into TF-IDF vectors (using up to 1000 features and excluding English stop words), and apply KMeans clustering to group samples based on textual similarity. The resulting clusters are treated as synthetic categories, and client-specific datasets are created by applying a Dirichlet-based allocation to these clusters. Additionally, we consider a more heterogeneous non-i.i.d. scenario in which each client is assigned data from a distinct task domain.

We set the number of segments N_s to 5 and set the sparsity rates as $k_{\max} = 0.95$, $k^A_{\min} = 0.6$, and $k^B_{\min} = 0.5$. We apply LoRA only to the self-attention layers, following (Hu et al., 2022). For QA tasks, in accordance with (Zhang et al., 2024; Wang et al., 2024), we set the rank r to 16, the scaling factor α to 32, and use a learning rate of 3×10^{-4}. For VA tasks, following (Ye et al., 2024), we choose $r = 8$, $\alpha = 16$, and a learning rate of 5×10^{-4}. For the Vicuna-7B model, we use an uncensored instruction-following model trained on the filtered WizardLM dataset (Xu et al., 2023), which does not incorporate human-aligned values. All datasets and models are used strictly for research purposes, in accordance with their respec-

tive licenses. When counting the total communication parameters, we exclude those required to distribute the initial pre-trained LLM. To measure communication time, we repeat each experiment five times and report the average. Experiments on Llama2-7B are conducted using two NVIDIA GeForce RTX 4090 GPUs, while those on Llama2-13B use an NVIDIA H100 GPU.

B Convergence Proof

We analyze the convergence of our method following the standard framework adopted in FL literature (Li et al., 2019). We assume that the global objective function F is differentiable and L-smooth (i.e., its gradient is L-Lipschitz continuous).

In each communication round t, the global model is updated as:

$$P_{t+1} = P_t - \eta U_t,$$

with the effective update given by:

$$U_t = \nabla F(P_t) + E_t,$$

where E_t contains errors from compression and round-robin segmentation. By the L-smoothness of F, we have:

$$F(P_{t+1}) \leq F(P_t) + \langle \nabla F(P_t), P_{t+1} - P_t \rangle + \frac{L}{2}\|P_{t+1} - P_t\|^2.$$

By substituting:

$$P_{t+1} - P_t = -\eta \left(\nabla F(P_t) + E_t \right),$$

we get:

$$F(P_{t+1}) \leq F(P_t) - \left(\eta - \frac{L\eta^2}{2}\right) \|\nabla F(P_t)\|^2 \underbrace{- \left(\eta - L\eta^2\right) \langle \nabla F(P_t), E_t \rangle}_{\triangleq A} + \frac{L\eta^2}{2}\|E_t\|^2. \tag{8}$$

Then, using the identity

$$\langle a, b \rangle = \frac{1}{2}\left(\|a\|^2 + \|b\|^2 - \|a-b\|^2 \right),$$

we have:

$$\begin{aligned} A &= -\eta(1-\eta L)\langle \nabla F(P_t), E_t \rangle \\ &= -\frac{\eta}{2}(1-\eta L)\|\nabla F(P_t)\|^2 - \frac{\eta}{2}(1-\eta L)\|E_t\|^2 \\ &\quad + \frac{\eta}{2}(1-\eta L)\|\nabla F(P_t) - E_t\|^2. \end{aligned}$$

Substituting back into the inequality (8):

$$\begin{aligned} F(P_{t+1}) \leq F(P_t) &- \eta \left(\frac{3}{2} - \eta L\right) \|\nabla F(P_t)\|^2 \\ &+ \eta \left(\eta L - \frac{1}{2}\right) \|E_t\|^2 \\ &+ \frac{\eta}{2}(1-\eta L)\|\nabla F(P_t) - E_t\|^2 \end{aligned}$$

Assume $\frac{\eta}{2}(1-\eta L) < 0 \Rightarrow \eta > \frac{1}{L}$, we have:

$$\begin{aligned} F(P_{t+1}) \leq F(P_t) &- \eta \left(\frac{3}{2} - \eta L\right) \|\nabla F(P_t)\|^2 \\ &+ \eta \left(\eta L - \frac{1}{2}\right) \|E_t\|^2 \end{aligned} \tag{9}$$

Now, we can decompose the error term E_t as:

$$E_t = E_t^{\text{comp}} + E_t^{\text{segment}},$$

where E_t^{comp} denotes the adaptive compression error, and E_t^{segment} denotes the segment sharing error. We denote the adaptive sparsification operator as $C(\cdot)$, which satisfies a contractive property, that is, for any vector x, there exists a constant $\delta \in (0, 1]$ such that:

$$\|C(x) - x\|^2 \leq (1-\delta)\|x\|^2.$$

Then, we get the following bound on the error E_t^{comp}:

$$\|E_t^{\text{comp}}\|^2 \leq (1-\delta) \|\nabla F(P_t)\|^2 .$$

In our algorithm, each client updates only one segment per round. Thus, a specific segment only gets updated once every N_s rounds. We denote by P_t the current global parameters and P_τ the stale parameters from the last round a given client participated. Then by the L-smoothness property, we have:

$$\|\nabla F(P_t) - \nabla F(P_\tau)\| \leq L\|P_t - P_\tau\|.$$

Since the change in parameters over each round is on the order of the learning rate η times the gradient, which we assume is bounded by some G, we can get:

$$\|\nabla F(P_t) - \nabla F(P_\tau)\| \leq L\, \eta\, N_s\, G.$$

As our algorithm uses an exponential decay weighting when updating the local model, we have:

$$\|E_t^{\text{segment}}\|^2 \leq \sum_{j=1}^{N_s} e^{-\beta j} \cdot (L\eta N_s G)^2 .$$

Because the sum $\sum_{j=1}^{N_s} e^{-\beta j}$ is a geometric series that converges to $\frac{e^{-\beta}}{1-e^{-\beta}}$, we obtain a bound of the form:

$$\|E_t^{\text{segment}}\|^2 \leq \frac{e^{-\beta}}{1-e^{-\beta}} L^2 \eta^2 N_s^2 G^2.$$

We define $\Delta = \frac{e^{-\beta}}{1-e^{-\beta}} L^2 \eta^2 N_s^2 G^2$, we have:

$$\begin{aligned}\|E_t\|^2 &\leq 2\|E_t^{\text{comp}}\|^2 + 2\|E_t^{\text{segment}}\|^2 \\ &= 2(1-\delta)\|\nabla F(P_t)\|^2 + 2\Delta\end{aligned}$$

Substituting into the inequality (9):

$$\begin{aligned}&F(P_{t+1}) \leq F(P_t) + \eta\,(2\eta L - 1)\cdot\Delta \\ &- \eta\left(\frac{5}{2} + \delta(2\eta L - 1) - 3\eta L\right)\|\nabla F(P_t)\|^2\end{aligned}$$

We define $\mu = \eta(\frac{5}{2} + \delta(2\eta L - 1) - 3\eta L)$, then:

$$\mu\|\nabla F(P_t)\|^2 \leq F(P_t) - F(P_{t+1}) + \eta\,(2\eta L - 1)\cdot\Delta$$

Summing both sides over $t = 0$ to $T-1$:

$$\sum_{t=0}^{T-1} \mu\|\nabla F(P_t)\|^2 \leq F(P_0) - F^* + T\eta\,(2\eta L - 1)\cdot\Delta$$

Finally, assuming $\mu > 0 \Rightarrow \eta < \frac{5-2\delta}{(6-4\delta)L}$, we have:

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla F(P_t)\|^2 \leq \frac{F(P_0) - F^*}{\mu T} + \frac{\eta\,(2\eta L - 1)\,\Delta}{\mu}$$

Choosing $\eta = O(\frac{1}{\sqrt{T}})$ ensures the average squared gradient norm decays as:

$$\frac{1}{T}\sum_{t=0}^{T-1} |\nabla F(P_t)|^2 = O\left(\frac{1}{\sqrt{T}}\right).$$

This completes the convergence proof.

C Additional Ablation Study

Comparison with top-k sparsification. Our proposed adaptive sparsification method exploits the differing sparsity patterns of matrices A and B throughout the training process, in contrast to the fixed threshold used in standard Top-k sparsification. In this section, we present a detailed comparison between the two approaches under varying compression levels. Specifically, we vary the threshold k for Top-k sparsification while ensuring that our adaptive sparsification uses the same total communication budget. The results are shown in Table 5. As shown, while Top-k sparsification achieves comparable performance to our method under low compression, it suffers from performance degradation as the compression level increases. This drop is primarily due to its inability to adapt to the evolving training dynamics and heterogeneous parameter patterns.

Table 5: Comparison of ARC of Top-k and Adaptive Sparsification under varying compression levels.

Threshold k	**Fixed Top-k**	**Adaptive Sparsification**
0.9	66.5	66.6
0.7	66.1	66.5
0.6	66.1	66.5
0.5	65.8	66.3

Experiments under Non-IID Conditions with Task Heterogeneity. In some extreme federated learning scenarios, each client may possess a significantly different data distribution, such as having a distinct task domain. It is important to assess the performance of EcoLoRA under such heterogeneous conditions. We evaluate our method on the Databricks-Dolly-15k dataset by assigning each client a unique task type based on the dataset's category field, using LLaMA-7B as the base model. The results are shown in Table 6. As shown, EcoLoRA achieves substantial reductions in communication overhead while maintaining competitive performance across non-IID, task-diverse clients.

Table 6: Comparison of accuracy and parameters (in millions) under non-IID conditions divided by task domain.

Method	**ARC**	**Upload Param.**	**Total Param.**
FedIT	0.664	2348.8	4697.6
FedIT w/ EcoLoRA	0.664	285.5	2157.3
FLoRA	0.663	2181.0	23991.4
FLoRA w/ EcoLoRA	0.663	292.5	19105.3
FFA-LoRA	0.665	1090.5	2181.0
FFA-LoRA w/ EcoLoRA	0.666	136.8	995.0