Global Adaptive Momentum Meets Local Personalized Perturbation: Efficient Federated LLM Fine-Tuning with Zeroth-Order Gradients

Zihan Chen[1], **Howard H. Yang**[2], **Tony Q.S. Quek**[1], **and Kai Fong Ernest Chong**[1*]
[1]Singapore University of Technology and Design
[2]ZJU-UIUC Institute, Zhejiang University
zihan_chen@sutd.edu.sg

Abstract

Federated fine-tuning of large language models (LLMs) provides a privacy-preserving approach to deploying pervasive generative AI services, yet the substantial memory overhead of first-order (FO) gradient computation presents significant practical challenges. While zeroth-order (ZO) optimization methods offer memory-efficient alternatives, they remain susceptible to performance degradation brought by data heterogeneity. Specifically, direct ZO-for-FO substitution is incompatible with existing strategies tailored for cross-client discrepancies. In response, we propose a new federated LLM fine-tuning framework, with a *holistic revamped design* of the entire ZO gradient processing pipeline. Crucially, with our proposed global adaptive optimization and local personalized perturbation, we present a unified solution for incorporating ZO gradients in federated learning, from local personalized perturbation sampling and ZO gradient transmission, to global ZO gradient reconstruction and aggregation with adaptive momentum, thereby directly addressing the challenges of inefficiencies and cross-client discrepancies. Our convergence analysis and experimental results demonstrate the superiority of our proposed framework over diverse heterogeneous data settings, both in terms of generalization and efficiency.

1 Introduction

Fine-tuning pre-trained large language models (LLMs) is essential for numerous downstream generative AI services (Bommasani et al., 2023; Achiam et al., 2023; Touvron et al., 2023). However, current pre-trained LLMs have largely exhausted the available Internet-scale training data, making personal and privately-held data the next critical resource frontier. Such personal data are inherently distributed across networks and subject to stringent privacy constraints, making centralized data collection impractical (Sani et al., 2025; Ye et al., 2024b). Conversely, fine-tuning exclusively on isolated private datasets may not achieve desirable performance. In this context, integrating federated learning (FL) with LLM fine-tuning emerges as a promising paradigm, facilitating effective utilization of distributed data for model fine-tuning while preserving data privacy (Kuang et al., 2024; Chen et al., 2023; Wang et al., 2024; Wu et al., 2024a).

As a practical solution for distributed learning implementations, FL enables multiple clients to train a machine learning model without sharing data, under the coordination of a central server (McMahan et al., Apr. 2017; Bonawitz et al., 2019). Typically, the significant *communication* overhead and data heterogeneity would significantly hinder the performance when deploying practical FL systems (Li et al., 2019; Wang et al., 2020). Such issues are exacerbated in the context of federated LLM fine-tuning (Woisetschläger et al., 2024). On the other hand, full-model fine-tuning of LLMs at local clients with first-order (FO) optimization methods, especially on edge devices with limited memory and computation budget, would introduce substantial *computation* and *memory* overhead, thereby significantly impeding system scalability (Fang et al., 2022; Ling et al., 2024). Given the challenges posed by the cost-intensive nature of LLMs, there is a critical need for efficient federated LLM fine-tuning in terms of multiple factors: communication, computation, and memory. Accordingly, diverse computation-efficient methods have been proposed to mitigate the computational overhead in federated LLM fine-tuning, such as parameter-efficient fine-tuning (PEFT) and its variants with FO optimization (Wang et al., 2024; Chen et al., 2024b; Guo et al., 2025a). However, such methods still do not perform as well as full-model tuning, and the number of exchangeable parameters (e.g., in LoRA (Hu et al., 2022)) still increases

*Corresponding Author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 39861–39875

proportionally as model size scales up, leading to large communication overhead in FL.

To tackle these challenges, numerous works have focused on adopting zeroth-order (ZO) optimization during local updates in FL to achieve full-model tuning with limited computation budget, in which local clients only need to perform forward passes to compute gradients at the cost of inference-time memory overhead (Fang et al., 2022; Jiang et al., 2024; Malladi et al., 2023; Qin et al., 2024). ZO methods utilize finite differences of loss function queries to estimate FO gradients for further descent-based model updates, bypassing the back-propagation process, and hence leading to significantly reduced memory overhead. Nevertheless, ZO federated LLM fine-tuning also suffers from unstable convergence and low efficiency in function queries. A key underlying factor is that simply replacing FO gradients with ZO gradients would negate various federated optimization techniques that are tailored to address cross-client discrepancies (Fang et al., 2022; Ling et al., 2024). This necessitates a balanced solution that addresses local inefficiencies in ZO optimization while achieving robust convergence performance. Furthermore, the random perturbation-based local ZO gradient computations would also inevitably face the challenge of divergent local gradients due to heterogeneous data, leading to degraded performance.

Hence, to fully leverage the potential of ZO gradients and address the above concerns, we propose a holistic redesign of the gradient processing pipeline, to obtain a federated LLM fine-tuning framework that is efficient across all factors: communication, computation, and memory. In particular, we aim to enhance the convergence and generalization performance of federated LLM fine-tuning with adaptive momentum training techniques for ZO gradients. The key idea is to design a robust global adaptive optimization method for model updates at the server, while still keeping local ZO model tuning. The proposed method could achieve full-model fine-tuning with a limited local computation budget, while obtaining accelerated and stabilized convergence performance. In addition, we propose a personalized perturbation scheme to improve the efficiency of federated ZO optimization, which has the net effect that the dimension of the client perturbation vector subspace increases at a faster rate. To address communication overhead, a lightweight "ZO information" update mechanism is also adopted to achieve $\mathcal{O}(1)$ communication cost in the client-to-server link (Qin et al., 2024).

Our contributions can be summarized as follows:

- We propose a new efficient federated LLM tuning framework with ZO optimization, whereby a holistic revamped design of the entire ZO gradient processing pipeline is introduced. In particular, we leverage global adaptive optimization and local personalized perturbation to tackle the performance degradation and inefficiencies of ZO FL with heterogeneous data.
- Our proposed personalized perturbation, as compared to previous approaches, allows for a much higher dimension for the overall span of the client perturbation vectors. Together with our global adaptive momentum, the overall framework yields better convergence performance while maintaining model-agnostic communication cost.
- Convergence analysis and extensive experiment results demonstrate the outperformance of our proposed method. We show that our proposed method has superior generalization and efficiency performance over diverse language tasks and heterogeneous data settings.

2 Related Work

Efficient federated LLM tuning. Recent approaches to efficient LLM tuning have focused on PEFT methods (e.g., LoRA (Hu et al., 2022), QLoRA (Dettmers et al., 2023), and prefix/prompting tuning (Li and Liang, 2021; Zhao et al., 2023)), sparsification (Guo et al., 2024), and quantization techniques (e.g., AWQ (Lin et al., 2024)), that reduce computational and memory requirements while maintaining desirable performance. Building upon these advancements in efficient LLM fine-tuning, recent efforts have extended these approaches to the federated setting (Ye et al., 2024a). Guo et al. (2025a); Zhang et al. (2023); Sun et al. (2024b) adapt PEFT techniques to the federated setting, demonstrating how PEFT-based methods can significantly reduce communication costs while maintaining performance in FL, where in particular, LoRA and its variants in FL are widely explored (Wang et al., 2024; Wu et al., 2024b; Sun et al., 2024b; Zhao et al., 2025a). For example, FLoRA (Wang et al., 2024) explores heterogeneous LoRA aggregation over heterogeneous networks. Guo et al. (2025a) propose selective

aggregation of LoRA matrices by leveraging the different roles of low-rank matrices. Furthermore, other techniques for reducing communication and memory costs in federated LLM fine-tuning are also widely investigated, such as adapter (Wu et al., 2024a), prompt tuning (Lv et al., 2024; Zhao et al., 2023), and quantization (Hadish et al., 2024).

ZO optimization in LLM fine-tuning and FL. ZO optimization has emerged as an effective approach for scenarios where gradients are difficult to compute or inaccessible (Liu et al., 2020; Jiang et al., 2024). DeepZero establishes the viability of scaling ZO optimization techniques to deep neural network training from scratch based on coordinate-wise gradient estimation (CGE) (Chen et al., 2024a). In LLM tuning, ZO methods circumvent the computational burden of backpropagation through massive parameter spaces (Jiang et al., 2024; Zhang et al., 2024; Guo et al., 2025b). MeZO achieves comparable performance to fine-tuning LLMs by adapting ZO-SGD, with the same memory footprints as inference (Malladi et al., 2023). HiZOO leverages diagonal Hessians to enhance ZO LLM fine-tuning with improved convergence and reduced memory costs (Zhao et al., 2025b). In the context of distributed scenarios, FedZO proposed a communication-efficient framework that enables collaborative model training through ZO optimization-based local updates (Fang et al., 2022). Additionally, to address the communication overhead in federated LLM fine-tuning, a scalar-type ZO gradient transmission scheme is proposed to achieve extremely low communication costs (Li et al., 2025). These advancements highlight the versatility of ZO methods in addressing both computational efficiency in LLM training and privacy preservation.

Overall, existing methods generally focus on either tuning a subset of LLM parameters via PEFT and its variants to enhance computation efficiency, or leveraging ZO optimization to improve memory efficiency in local tuning or reduce communication overhead in FL. Our proposed method proposes a holistic framework for federated LLM fine-tuning with ZO optimization, in which we revamp the pipeline of ZO gradient processing to improve the efficiency while maintaining decent performance with data heterogeneity.

3 Preliminaries

3.1 FO optimization methods in FL

We consider an FL system with N clients. Within round t, the typical training pipeline of conventional FO optimization-based FL (McMahan et al., Apr. 2017) is described as follows. First, each selected i-th client with loss function $f_i : \mathbb{R}^d \to \mathbb{R}$ computes its local updates $\nabla f_i(\boldsymbol{w}^t)$ via FO optimization with gradient descent over the latest model weights $\boldsymbol{w}^t \in \mathbb{R}^d$, after which that client sends out the FO gradients to the server. After collecting all local FO gradients, the server performs aggregation to update gradients $\boldsymbol{g}^t$ via $\boldsymbol{g}^t = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i\left(\boldsymbol{w}^t\right)$. Then the server updates $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \eta \boldsymbol{g}^t$ and broadcasts the updated model for the next round of computations, where η is the learning rate. For simplicity, we use $\nabla f_i(\boldsymbol{w}^t)$ to denote both FO gradients and multi-step model updates, depending on the context.

During local FO update, the backpropagation process in local training incurs substantial computational and memory overhead, especially when recent models have billions of parameters (Touvron et al., 2023). Hence, fine-tuning LLMs frequently exceeds the realistic computation and memory limits of resource-constrained clients over heterogeneous networks. The transmission of full-model gradients with full model size also brings huge communication costs. Directly adopting PEFT into FL could alleviate these concerns. Taking LoRA as an example (Wang et al., 2024; Hu et al., 2022), the local computation and parameter aggregation are built upon the (updated) weights of low-rank matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ for client i. The updated process is given by: $\boldsymbol{A} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{A}_i, \boldsymbol{B} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{B}_i$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are updated LoRA matrices for the next round of computations. However, such a process would still require backpropagation for low-rank matrix updates, which incurs significant cumulative communication overhead.

3.2 ZO optimization

ZO optimization bypasses the backpropagation process for gradient computation, thereby vastly reducing the memory and computation overhead. The usual goal of ZO optimization is to approximate the FO gradient of a loss function $f(\boldsymbol{w})$, where $\boldsymbol{w} \in \mathbb{R}^d$ denotes model weights, based solely on the finite differences of loss values $f(\boldsymbol{w}')$ at multiple different realizations of $\boldsymbol{w}'$. Specifically, two prominent gradient estimation schemes are used to

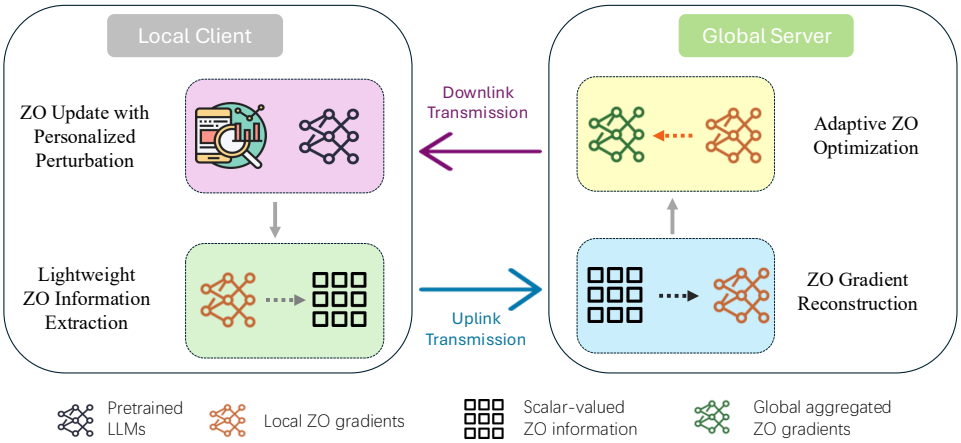

Figure 1: A depiction of our proposed framework, where the steps within a single communication round are shown.

calculate ZO gradient: randomized gradient estimation (RGE) (Malladi et al., 2023; Zhang et al., 2024) and CGE (Chen et al., 2024a). Although CGE[1] typically demonstrates superior performance, RGE provides a more computationally efficient alternative (Chen et al., 2024a). In this paper, we focus on the RGE-based ZO gradient estimation. Given a reference point $\boldsymbol{w} \in \mathbb{R}^d$, the RGE-based gradient estimate $\widehat{\nabla} f(\boldsymbol{w})$ is computed as the average of multiple intermediate estimates:

$$\widehat{\nabla} f(\boldsymbol{w}) = \frac{1}{q} \sum_{j=1}^{q} \left(f(\boldsymbol{w} + \mu \boldsymbol{z}_j) - f(\boldsymbol{w}) \right) \frac{\boldsymbol{z}_j}{\mu}, \quad (1)$$

where q is the number of perturbation vectors (i.e., number of function queries), $\boldsymbol{z}_j \in \mathbb{R}^d$ is the j-th random perturbation vector ($1 \leq j \leq q$) sampled from the standard Gaussian distribution $\mathcal{N}(0, \boldsymbol{I}_d)$, and $\mu > 0$ is the smoothing step-size. As such, only forward passes are needed to compute $\widehat{\nabla} f(\boldsymbol{w})$ (Malladi et al., 2023; Qin et al., 2024; Li et al., 2025). Here, $\widehat{\nabla} f(\boldsymbol{w})$ is an efficient approximation of the FO gradient $\nabla f(\boldsymbol{w})$, which allows for obtaining gradient approximation with limited computation budget. In particular, FedZO makes the first attempt to apply ZO into local updates in FL, in which each client updates its own local ZO gradients for global aggregation (Fang et al., 2022). However, existing ZO FL methods do not directly address the data heterogeneity across clients.

4 Proposed Framework

Motivated by the above insights, we aim to explore a holistic ZO gradient processing pipeline for efficient federated LLM fine-tuning, to tackle the issues of inefficiency and data heterogeneity. Instead of solely replacing FO gradients with ZO gradients in FL, we take a more comprehensive approach for ZO optimization for FL, by considering all steps involving ZO gradients, from random perturbation sampling, local ZO gradient estimation and transmission, to global ZO gradient aggregation and update; see Fig. 1 for an illustration. In particular, our framework consists of two major components: local ZO update with personalized perturbation, and global adaptive optimization. In the subsequent subsections, we shall systematically introduce the individual steps of our entire ZO gradient processing pipeline, covering personalized perturbation sampling, ZO local updates, lightweight transmission, global ZO gradient reconstruction, and adaptive momentum-based model updates.

4.1 Local ZO update with personalized perturbation

Personalized local perturbation. We first introduce personalized local perturbations for local ZO updates. Existing works on federated LLM fine-tuning with ZO gradients mainly focus on efficiency but ignore the discrepancies across clients with heterogeneous data. In contrast, we propose a normalized perturbation technique to align all random perturbation vectors across different ZO estimation steps and clients. For a given sampled perturbation vector $\boldsymbol{z}_j = \boldsymbol{z}_j(s_j)$, sampled from $\mathcal{N}(0, \boldsymbol{I}_d)$ using random seed s_j, we first compute its normalization $\tilde{\boldsymbol{z}}_j = \tilde{\boldsymbol{z}}_j(s_j)$ by:

$$\tilde{\boldsymbol{z}}_j := \boldsymbol{z}_j / ||\boldsymbol{z}_j||^2. \quad (2)$$

[1]Informally, CGE uses *all* standard basis vectors for perturbation, while RGE uses *some* random perturbation vectors, which span a much smaller subspace.

Next, for any given $\boldsymbol{w} \in \mathbb{R}^d$, we propose a revised RGE-based ZO gradient estimate:

$$\widetilde{\nabla} f(\boldsymbol{w}) = \frac{1}{q} \sum_{j=1}^{q} \left(f(\boldsymbol{w} + \mu \tilde{\boldsymbol{z}}_j) - f(\boldsymbol{w}) \right) \frac{\tilde{\boldsymbol{z}}_i}{\mu}, \quad (3)$$

where q is the number of perturbation vectors, and $\mu > 0$ is the smoothing step-size. Hence, within round t, the detailed local training steps for client i can be reformulated as:

$$\boldsymbol{w}_{i,k+1}^t \leftarrow \boldsymbol{w}_{i,k}^t - \eta \widetilde{\nabla} f(\boldsymbol{w}_{i,k}^t), \quad (4)$$

where E is the total number of local ZO optimization steps, $k = 0, \ldots, E-1$, and $\boldsymbol{w}_{i,0}^t = w^t$ is the latest global model weights in round t, and $\widetilde{\nabla} f(\boldsymbol{w}_{i,k}^t)$ is computed via Eq. 3.

Note that RGE-based gradient estimation methods enjoy better computational efficiency but suffer from low representation capabilities for high-dimensional subspaces as compared to CGE-based methods, whereby there is a trade-off between computation efficiency and expressiveness. Intuitively, for faster model convergence, gradient updates should not be restricted to low-dimensional subspaces. Hence, perturbation vectors across different clients and different steps should be as diverse as possible, so that the dimension of the span of perturbation vectors is maximized. To achieve this while maintaining $\mathcal{O}(1)$ communication cost in the client-to-server link, we propose a simple yet effective approach: Within each round, different clients use different random seed sequences without repeats for perturbation vector sampling. See supplementary material for more details.

Lightweight ZO information update. In round t, after the E local ZO optimization steps have been completed at client i, the next task is to upload the information about the local ZO update $\boldsymbol{w}_{i,E}^t - \boldsymbol{w}^t$. This is a vector of length d, thus transmitting it directly to the server would incur significant communication costs. Crucially, $\boldsymbol{w}_{i,E}^t - \boldsymbol{w}^t$ is a linear combination of ZO gradients $\{\widetilde{\nabla} f(\boldsymbol{w}_{i,k}^t)\}_{k=1}^E$. This allows us to achieve $\mathcal{O}(1)$ cost for uplink transmission, by adapting the scalar-valued update scheme from (Qin et al., 2024; Li et al., 2025). In particular, each ZO gradient $\widetilde{\nabla} f(\boldsymbol{w}_{i,k}^t)$ is given by

$$\widetilde{\nabla} f(\boldsymbol{w}_{i,k}^t) = \frac{1}{q} \sum_{j=1}^{q} \Delta_{\{i,k\},j}^t \cdot \tilde{\boldsymbol{z}}_{\{i,k\},j}^t, \quad (5)$$

in which $\Delta_{\{i,k\},j}^t$ is a scalar and could be denoted as $\Delta_{\{i,k\},j}^t := \frac{1}{\mu}(f(\boldsymbol{w}_{i,k}^t + \mu \tilde{\boldsymbol{z}}_{\{i,k\},j}^t) - f(\boldsymbol{w}))$, This means that $\widetilde{\nabla} f(\boldsymbol{w}_{i,k}^t)$ is completely determined by $\{\Delta_{\{i,k\},j}^t\}_{j=1}^q$ and the normalized perturbation vectors, where in turn, the normalized perturbation vectors are uniquely determined by a set of random seeds $\{s_{\{i,k\},j}^t\}_{j=1}^q$. Consequently, the server only needs a total of $q \cdot E$ pairs of scalars, i.e., $\{(\Delta_{\{i,k\},j}^t, s_{\{i,k\},j}^t)_{j=1}^q\}_{k=1}^E$, to reconstruct the corresponding estimated gradients for client i. This collection of scalar pairs shall henceforth be called *ZO information*. To summarize, the uplink communication cost is NqE, where N is total number of clients, q is the number of perturbation vectors, and E is the number of local training steps, which is independent of the model size d and satisfies $NqE \ll d$.

Algorithm 1 Efficient federated LLM fine-tuning

Inputs: N, T, $\boldsymbol{w}^1$ η, $\boldsymbol{m}^1$, $\boldsymbol{v}^1$, E
Outputs: Fine-tuned model w^{T+1}

for $t = 1$ **to** T **do**
  for each client $i = 1$ **to** N **in parallel do**
    // Zeroth-order model fine-tuning
    Obtain $\{(\Delta_{\{i,k\},j}^t, s_{\{i,k\},j}^t)_{j=1}^q\}_{k=1}^E$ via performing LocalZOUpdate
    Upload scalar pairs to server
    // Global ZO gradient processing
    Reconstruct ZO gradients via Eq. 6
    Compute $\hat{\boldsymbol{g}}^t$ via adaptive momentum method (Eq. 7 to Eq. 9)
    $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \eta \hat{\boldsymbol{g}}^t$
return $\boldsymbol{w}_{\mathrm{G}}^T$

function LocalZOUpdate($\boldsymbol{w}^t, \{s_{\{i,k\},j}^t\}$)
  Require: $\boldsymbol{w}^t, \{s_{\{i,k\},j}^t\}$
  $w_{i,0}^t \leftarrow w^t$
  for $j = 1$ **to** E **do**
    // Local ZO update
    $\boldsymbol{w}_{i,k+1}^t \leftarrow \boldsymbol{w}_{i,k}^t - \eta \widehat{\nabla} f(\boldsymbol{w}_{i,k}^t)$
    Obtain local ZO scalar pairs $\{(\Delta_{\{i,k\},j}^t, s_{\{i,k\},j}^t)_{j=1}^q\}_{k=1}^E$
  return $\{(\Delta_{\{i,k\},j}^t, s_{\{i,k\},j}^t)_{j=1}^q\}_{k=1}^E$

4.2 Global adaptive optimization

Given the heterogeneous ZO updates, we propose global adaptive optimization techniques to stabilize the training and improve the convergence performance. The global adaptive optimization consists of two components: ZO gradient reconstruction and adaptive ZO gradient updates, where the goal of the adaptive optimization is not only to address

the divergence brought by data heterogeneity, but also to deal with the variance induced by the random perturbation sampling.

ZO gradient reconstruction. Upon receiving the scalar-valued local ZO updates from client i, the server needs to perform ZO gradient reconstruction first, by computing

$$\widetilde{\nabla} f_i(\boldsymbol{w}_i^t) = \frac{1}{q} \sum_{k=1}^{E} \sum_{j=1}^{q} \Delta_{\{i,k\},j}^t \frac{\tilde{\boldsymbol{z}}_i(s_{\{i,k\},j}^t)}{\mu}, \quad (6)$$

after which the global ZO gradient can be obtained via aggregation $\boldsymbol{g}^t \leftarrow \frac{1}{N} \sum_{i=1}^{N} \widetilde{\nabla} f_i\left(\boldsymbol{w}^t\right)$.

ZO gradient adaptive optimization. Next, the server shall obtain aggregated ZO gradients $\boldsymbol{g}^t$ for the subsequent adaptive optimization. Specifically, we introduce intermediate momentum-type vectors $\boldsymbol{m}^t$ and $\boldsymbol{v}^t$. Adopting adaptive momentum and learning rate with exponential moving average, the adaptive ZO updates are computed as follows:

$$\boldsymbol{m}^t \leftarrow \beta_1 \boldsymbol{m}^{t-1} + (1-\beta_1)\boldsymbol{g}^t; \quad (7)$$

and for each ℓ-th entry ($1 \leq \ell \leq d$), we have

$$\boldsymbol{v}^t[\ell] \leftarrow \beta_2 \boldsymbol{v}^{t-1}[\ell] + (1-\beta_2)\boldsymbol{g}^t[\ell] \cdot \boldsymbol{g}^t[\ell], ; \quad (8)$$

$$\hat{\boldsymbol{g}}^t[\ell] \leftarrow \frac{\boldsymbol{m}^t[\ell]}{\sqrt{\boldsymbol{v}^t[\ell] + \varepsilon}}, . \quad (9)$$

Here, $0 \leq \beta_1, \beta_2 < 1$ are hyperparameters to control the momentum, ε is an arbitrary small positive number, and for any vector $\boldsymbol{a} \in \mathbb{R}^d$, we let $\boldsymbol{a}[\ell]$ denote its ℓ-th entry. After that, the server could choose to broadcast the updated gradients or the updated full model to all the local clients for the next round of local LLM fine-tuning. The new global model is given by $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \eta \hat{\boldsymbol{g}}^t$. Upon finishing a total of T rounds of training, we obtain $\boldsymbol{w}^{T+1}$. We also investigate variants for our proposed framework. More details can be found in Sec. B.2 of the supplementary material.

4.3 Algorithm summary

Building upon our revamped ZO gradient processing pipeline, we summarize our proposed efficient ZO federated LLM fine-tuning framework as Algorithm 1. As an overview, we begin within every round t with the server broadcasting the latest model to each client for local fine-tuning. Each client i performs E steps of local ZO optimization with personalized perturbation, followed by lightweight scalar-valued ZO information transmission. After collecting all local ZO updates, the server first reconstructs the ZO gradients for model aggregation with adaptive momentum. The server broadcasts the updated global to all clients, which marks the start of the next round $t+1$.

Discussion on the computation and communication overhead. Our proposed method keeps native ZO optimization for local updates while allowing for cost-intensive optimization at the server to address the mentioned challenges. Hence, the introduced computational overhead mainly arises at the server, and the peak memory costs arise at inference-time. By design, our framework incurs only $\mathcal{O}(1)$ uplink overhead while incurring $\mathcal{O}(d)$ downlink overhead, thereby matching the inherent asymmetric properties of real-world networks where downlink throughput far outstrips uplink. A detailed comparison can be found in Sec. 5.2.3.

4.4 Convergence analysis

We provide the convergence analysis for our proposed framework. We assume that the objective function $f(\cdot)$ is a lower bound defined by a minimum possible value f^* of $f(\boldsymbol{w})$ for all $\boldsymbol{w} \in \mathbb{R}^d$. To facilitate analysis, we adopt the following assumptions, commonly adopted in studies on FL and ZO optimization (Li et al., 2019, 2025).

Assumption 1. *(L-smooth) Every loss function $f_i(\cdot)$ is differentiable with domain $\mathbb{R}^d$ and L-smooth (for some fixed $L > 0$), i.e., for all $\boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^d$, we have $\|\nabla f_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{v})\| \leq L\|\boldsymbol{w} - \boldsymbol{v}\|$.*

Assumption 2. *(Unbiased and Bounded Stochastic Gradient) The local stochastic gradient is unbiased, i.e. $\mathbb{E}\left[\mathbb{E}\left[f_i(\boldsymbol{w}^t, \zeta_i^t)\right]\right] = \mathbb{E}\left[\nabla f_i(\boldsymbol{w}^t)\right]$, and $\mathbb{E}\left[\left\|\nabla f_i\left(\boldsymbol{w}^t; \zeta_t\right) - \nabla f_i(\boldsymbol{w}^t)\right\|^2\right] \leq \sigma_b^2$, for all i, for any $t \geq 1$ where $\zeta_i^{(t)}$ denotes the gradient noise for client i in round t.*

Assumption 3. *(Bounded Local Gradients and Similarity) We assume that the unbiased stochastic gradients have bounded second moments, i.e., there is a common constant $G > 0$ such that $\mathbb{E}\left[\left\|\nabla f_i\left(\boldsymbol{w}^t\right)\right\|^2\right] \leq G^2$, and for all i.*

We now present the convergence bound. A full proof can be found in the supplementary material.

Theorem 1. *Under Assumptions 1-3, and for possibly non-convex loss functions, with a total of T*

Table 1: Average (3 trials) and standard deviation of the best performance of different methods for fine-tuning OPT 1.3B and LLaMA 3B with heterogeneous data settings. The highest performance for each task is boldfaced.

Models	Methods	SST-2	SST-5	MNLI	RTE	QNLI
OPT 1.3B	FedAvg	$85.23_{\pm 0.95}$	$41.13_{\pm 1.42}$	$79.37_{\pm 0.85}$	$64.18_{\pm 0.83}$	$77.52_{\pm 0.53}$
	FFA LoRA	$83.35_{\pm 0.78}$	$41.35_{\pm 0.98}$	$78.63_{\pm 0.91}$	$64.25_{\pm 0.63}$	$76.35_{\pm 0.68}$
	FedMeZO	$82.59_{\pm 1.46}$	$36.57_{\pm 1.56}$	$75.34_{\pm 1.79}$	$62.13_{\pm 1.67}$	$73.68_{\pm 0.97}$
	FedZO	$82.93_{\pm 1.52}$	$38.26_{\pm 1.31}$	$77.49_{\pm 0.48}$	$63.89_{\pm 1.39}$	$74.83_{\pm 1.05}$
	FedKSeed	$83.43_{\pm 0.85}$	$40.67_{\pm 0.55}$	$78.21_{\pm 0.62}$	$63.94_{\pm 1.71}$	$76.52_{\pm 0.46}$
	DeComFL	$82.14_{\pm 1.37}$	$40.38_{\pm 0.87}$	$77.78_{\pm 0.43}$	$63.27_{\pm 0.76}$	$76.01_{\pm 0.75}$
	Ours	$\mathbf{84.02_{\pm 0.82}}$	$\mathbf{41.56_{\pm 0.62}}$	$\mathbf{78.70_{\pm 0.73}}$	$\mathbf{64.32_{\pm 0.95}}$	$\mathbf{77.29_{\pm 0.48}}$
LLaMA 3B	FedAvg	$87.36_{\pm 0.76}$	$43.58_{\pm 1.13}$	$80.97_{\pm 0.34}$	$66.92_{\pm 0.72}$	$80.14_{\pm 0.69}$
	FFA LoRA	$85.21_{\pm 0.52}$	$43.69_{\pm 0.96}$	$\mathbf{81.34_{\pm 0.29}}$	$67.39_{\pm 0.41}$	$79.88_{\pm 0.31}$
	FedMeZO	$84.63_{\pm 1.75}$	$39.20_{\pm 1.57}$	$77.68_{\pm 0.96}$	$64.51_{\pm 1.24}$	$75.32_{\pm 1.39}$
	FedZO	$84.79_{\pm 1.81}$	$40.91_{\pm 1.72}$	$79.14_{\pm 0.72}$	$65.86_{\pm 1.17}$	$79.17_{\pm 0.74}$
	FedKSeed	$85.86_{\pm 0.55}$	$43.25_{\pm 0.42}$	$80.96_{\pm 0.32}$	$67.03_{\pm 0.93}$	$80.52_{\pm 0.52}$
	DeComFL	$85.08_{\pm 0.93}$	$42.47_{\pm 1.05}$	$80.32_{\pm 0.69}$	$66.85_{\pm 1.05}$	$80.12_{\pm 0.66}$
	Ours	$\mathbf{86.32_{\pm 0.74}}$	$\mathbf{44.07_{\pm 0.58}}$	$81.27_{\pm 0.35}$	$\mathbf{67.40_{\pm 0.84}}$	$\mathbf{81.08_{\pm 0.54}}$

Table 2: Performance comparison of fine-tuning OPT 1.3B under diverse heterogeneous data settings.

Methods	SST-2		RTE	
	$\alpha = 10$	$\alpha = 0.1$	$\alpha = 10$	$\alpha = 0.1$
FedMeZO	83.07	81.95	63.27	61.86
FedZO	83.25	82.07	64.05	62.59
FedKSeed	83.86	82.56	64.78	63.12
DeComFL	83.13	81.38	64.52	62.91
Ours	**84.21**	**83.87**	**64.82**	**64.15**

rounds, Algorithm 1 converges as follows:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla f(\boldsymbol{w}^t)\|^2\right] \leq \frac{2\sqrt{G^2+\varepsilon}}{\eta T}(f(\boldsymbol{w}^1)-f^*) + \Gamma + \frac{L\eta G^2\sqrt{G^2+\varepsilon}}{\varepsilon}, \tag{10}$$

where $f^* = \min\{f(\boldsymbol{w})|\boldsymbol{w}\in\mathbb{R}^d\}$, $\Gamma = 8\kappa\beta_1^2G^2 + \frac{8c_d\kappa(\sigma_b^2+2(d+4)G^2)}{NEq} + \frac{4c_d\kappa\mu^2L^2(d+6)^3}{NEq} + \kappa\mu^2L^2(d+3)^3 + \frac{8\kappa L^2\eta^2G^2E^2}{3}$, *and* $\kappa = (G^2+\varepsilon)/\varepsilon$

From Theorem 1, we obtain the convergence rate of our proposed method. The two errors terms on the right-hand side shows the effects of the gradient estimate and heterogeneous data on the convergence. In particular, an increase in the number of perturbation vectors q, number of local steps E and total number of clients N would reduce the error term with improved convergence performance.

5 Experiments

5.1 Experimental setup

We evaluate our proposed method with diverse natural language processing tasks. Specifically, we adopt OPT 1.3B (Zhang et al., 2022) and LLaMA 3.2-3B (Touvron et al., 2023) as the pretrained model for federated fine-tuning tasks. For evaluation tasks, we use the GLUE benchmark (Wang et al., 2018), which includes SST-2, SST-5, MNLI, RTE, and QNLI. Regarding the federated system setup, unless otherwise stated, we used $N = 20$ for fine-tuning OPT 1.3B and used $N = 10$ for fine-tuning LLaMA 3.2-3B. To emulate heterogeneous data distributions, we used the symmetric Dirichlet distribution-based partition method with concentration parameter $\alpha = 0.5$, i.e., *Dir*(0.5) (Guo et al., 2025a; Qin et al., 2024). For our framework, we use $\mu = 10^{-3}$, $q = 50$, and $E = 5$ during ZO gradient estimation. Further implementation details can be found in the appendices.

Baselines. We compare our method with following state-of-the-art FL methods: ZO FL methods, including FedZO (Fang et al., 2022), FedKSeed (Qin et al., 2024), DeComFL (Li et al., 2025); and FFA LoRA (a federated PEFT method) (Sun et al., 2024a). We also implemente an FL version for the MeZO algorithm (Malladi et al., 2023), namely FedMeZO, in which local updates follow MeZO. For FO methods, we use the SGD optimizer. For ZO FL methods, we use the same number of perturbation vectors for fair

Table 3: Local peak memory overhead within each communication round of training of different methods of fine-tuning LLaMA 3.2-3B.

Methods	FedAvg	FFA LoRA	FedMeZO	FedZO	FedKSeed	DeComFL	Ours
Peak Memory Usage	39.1 GB	19.1 GB	7.8 GB	7.6 GB	7.8 GB	7.7 GB	7.9 GB

comparison. Furthermore, we provide the performance of FedAvg (McMahan et al., Apr. 2017); this is provided for reference only, since it yields full model fine-tuning and transmission with significant computation and communication overhead.

5.2 Performance evaluation

5.2.1 Performance comparison

We evaluate the generalization performance and efficiency performance of our proposed framework.

Generalization performance evaluation. We compared the best performance with multiple state-of-the-art baselines across diverse benchmarks with non-iid data, using the same system configuration. Tab. 1 gives the main results on OPT 1.3 and LLaMA 3.2-3B. In summary, our proposed framework achieves the best performance on almost all benchmarks, outperforming all FO federated PEFT and ZO FL baselines on both pretrained foundation models. Note that, given the massive parameter sizes in local update and transmission, these overheads make FedAvg impractical for the deployment of LLMs in real-world systems, we put it as a reference baseline only. In Tab. 2, we evaluate the performance of different ZO FL methods by fine-tuning OPT 1.3B over different non-iid settings, i.e., different α, which shows that our proposed framework demonstrates less performance variation when local data statistics vary. Overall, these results demonstrate that our proposed framework has consistently superior performance across diverse tasks, models, and heterogeneous data settings. Our proposed framework achieves better generalizability and stability, benefiting from the holistic revamped ZO gradient process pipeline.

Efficiency performance evaluation. We compare the memory efficiency of our proposed framework and baselines. In particular, we evaluate the peak memory usage within each round during local update. As shown in Tab. 3 (the values for FedAvg, FedZO, and FedKSeed in this table taken from (Qin et al., 2024)), our proposed method attains the desirable memory efficiency, while maintaining better generalization performance.

Table 4: Ablation study of our framework with fine-tuning OPT 1.3B on SST-2. LPP (resp. GAM) represents the local personalized perturbation (resp. global adaptive momentum) technique in our framework.

Setups	Ablation study		
	$N = 10$	$N = 20$	$N = 30$
Ours	84.07	84.02	84.45
Ours w/o LPP	83.59	83.84	83.77
Ours w/o GAM	83.52	83.65	83.60
Ours w/o both	82.85	83.13	83.52

Table 5: Sensitivity analysis of our framework with fine-tuning OPT 1.3B over diverse benchmarks.

Setups	Sensitivity analysis		
	$q = 10$	$q = 50$	$q = 100$
SST-2	83.51	84.02	84.29
SST-5	39.64	41.56	41.78
RTE	63.85	64.32	65.02
QNLI	76.91	77.29	77.31

5.2.2 Sensitivity analysis and ablation study

To further evaluate the efficiency of our proposed method, we conduct an ablation study as shown in Tab. 4. Specifically, we fine-tune the OPT 1.3B model over different system scales (i.e., different total number of clients N) with $q = 50$. It is demonstrated that both local personalized perturbation and global adaptive momentum can improve the performance, and increasing the system scale N leads to better performance for most setups, which aligns with the findings given in Theorem 1. In the right 4 columns of Tab. 5, we provide the sensitivity analysis for the number of perturbation vectors. Tab. 5 shows that having more perturbation vectors improves overall performance. This is consistent to the theoretical implications of Theorem 1. Further sensitivity analyses can be found in the appendices.

5.2.3 System level cost comparison

As shown in Tab. 6, we provide a system-level cost analysis to highlight the practical benefits of our method with regard to the uplink communica-

Table 6: A comparison of system-level cost (including uplink and downlink communication overhead and the computation overhead) of our method and other baselines.

Methods	Uplink	Downlink	Computation
FO Full FT + BP	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(Ad)$
PEFT + BP	$\mathcal{O}(\tau d)$	$\mathcal{O}(\tau d)$	$\mathcal{O}(Bd)$
ZO Full FT	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$
FedKSeed	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(d)$
Ours	$\mathcal{O}(1)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$

tion overhead, downlink communication overhead, and the computation overhead. In particular, d is the model size, τ is the ratio of PEFT's trainable parameters, A is the peak memory ratio for FO back-propagation (BP), and B is the peak memory ratio for PEFT with back-propagation (BP), in which $\tau \ll 1 < B < A$. Overall, our method achieves a good balance across all three system-level cost metrics while maintaining superior generalization and stability performance. Together with the above evaluation results, it is illustrated that our method has the potential to push the boundary of the efficiency-performance trade-off which balances computation and communication overhead for federated LLM fine-tuning with improved performance.

6 Conclusion

In this work, we proposed an efficient federated LLM fine-tuning framework based on ZO optimization to address the inefficiency and performance deterioration caused by heterogeneous data. In particular, we proposed a holistic revamped design of the entire ZO gradient processing pipeline, whereby global adaptive momentum and local personalized perturbation schemes are introduced. Our framework could effectively address the inherent issues of inefficiencies and cross-client discrepancies. As the broader impact, our framework provides a new, efficient solution for tuning and deploying LLMs to edge clients without sharing data for ubiquitous generative AI services. Furthermore, since our method is complementary to the downlink process, it can be seamlessly combined with existing SOTA compression or communication-efficient LLM transmission techniques.

Limitations

Our framework, as currently formulated, does not deal with the downlink communication overhead. Techniques dealing with both uplink and downlink communication overhead in this context would require further investigation.

Acknowledgement

This work is supported in part by the Ministry of Education, Singapore, under its Tier 2 Research Fund (MOE-T2EP20125-0011), in part by the SUTD Kickstarter Initiative (SKI 2021_03_01), and in part by the National Natural Science Foundation of China under Grant 62201504.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rishi Bommasani, Percy Liang, and Tony Lee. 2023. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*, 1525(1):140–146.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, and 1 others. 2019. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388.

Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Konstantinos Parasyris, Jiancheng Liu, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. 2024a. Deepzero: Scaling up zeroth-order optimization for deep model training. In *International Conference on Learning Representations*.

Haokun Chen, Yao Zhang, Denis Krompass, Jindong Gu, and Volker Tresp. 2024b. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11285–11293.

Zihan Chen, Howard H. Yang, Tony Q. S. Quek, and Kai Fong Ernest Chong. 2023. Spectral co-distillation for personalized federated learning. In *Advances in neural information processing systems*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Wenzhi Fang, Ziyi Yu, Yuning Jiang, Yuanming Shi, Colin N Jones, and Yong Zhou. 2022.

Communication-efficient stochastic zeroth-order optimization for federated learning. *IEEE Transactions on Signal Processing*, 70:5058–5073.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Saeed Ghadimi and Guanghui Lan. 2013. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368.

Jinyang Guo, Jianyu Wu, Zining Wang, Jiaheng Liu, Ge Yang, Yifu Ding, Ruihao Gong, Haotong Qin, and Xianglong Liu. 2024. Compressing large language models by joint sparsification and quantization. In *Forty-first International Conference on Machine Learning*.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. 2025a. Selective aggregation for low-rank adaptation in federated learning. In *The Thirteenth International Conference on Learning Representations*.

Wentao Guo, Jikai Long, Yimeng Zeng, Zirui Liu, Xinyu Yang, Yide Ran, Jacob R. Gardner, Osbert Bastani, Christopher De Sa, Xiaodong Yu, Beidi Chen, and Zhaozhuo Xu. 2025b. Zeroth-order fine-tuning of LLMs with transferable static sparsity. In *The Thirteenth International Conference on Learning Representations*.

Siem Hadish, Velibor Bojković, Moayad Aloqaily, and Mohsen Guizani. 2024. Language models at the edge: A survey on techniques, challenges, and applications. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pages 262–271. IEEE.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 1(2):3.

Shuoran Jiang, Qingcai Chen, Youcheng Pan, Yang Xiang, Yukang Lin, Xiangping Wu, Chuanyi Liu, and Xiaobao Song. 2024. Zo-adamu optimizer: Adapting perturbation by the momentum and uncertainty in zeroth-order optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18363–18371.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. 2024. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5260–5271.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Zhe Li, Bicheng Ying, Zidong Liu, Chaosheng Dong, and Haibo Yang. 2025. Achieving dimension-free communication in federated learning via zeroth-order optimization. In *International Conference on Learning Representations*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100.

Zhenqing Ling, Daoyuan Chen, Liuyi Yao, Yaliang Li, and Ying Shen. 2024. On the convergence of zeroth-order federated tuning for large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1827–1838.

Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. 2020. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54.

Guangtong Lv, Bruce Gu, Xiaocong Jia, Longxiang Gao, Youyang Qu, and Lei Cui. 2024. Federated learning and parallel prompt scheduling strategies for large language models. In *International Conference on Algorithms and Architectures for Parallel Processing*, pages 317–326. Springer.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. 2023. Fine-tuning language models with just forward passes. *Advances in neural information processing systems*, 36:53038–53075.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Apr. 2017. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, Fort Lauderdale, USA.

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 *Proceedings of the 41st International Conference on Machine Learning*, pages 41473–41497.

Lorenzo Sani, Alex Iacob, Zeyu Cao, Royson Lee, Bill Marino, Yan Gao, Wanru Zhao, Dongqi Cai, Zexi Li, Xinchi Qiu, and Nicholas D. Lane. 2025. Photon: Federated LLM pre-training. In *Eighth Conference on Machine Learning and Systems*.

Jingwei Sun, Ziyue Xu, Hongxu Yin, Dong Yang, Daguang Xu, Yudong Liu, Zhixu Du, Yiran Chen, and Holger R Roth. 2024a. Fedbpt: Efficient federated black-box prompt tuning for large language models. In *International Conference on Machine Learning*, pages 47159–47173. PMLR.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623.

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*.

Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. 2024. Federated fine-tuning of llms on the very edge: The good, the bad, the ugly. In *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning*, pages 39–50.

Feijie Wu, Zitao Li, Yaliang Li, Bolin Ding, and Jing Gao. 2024a. 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.

Panlong Wu, Kangshuo Li, Ting Wang, Yanjie Dong, Victor CM Leung, and Fangxin Wang. 2024b. Fedfmsl: Federated learning of foundations models with sparsely activated lora. *IEEE Transactions on Mobile Computing*.

Rui Ye, Rui Ge, Xinyu Zhu, Jingyi Chai, Du Yaxin, Yang Liu, Yanfeng Wang, and Siheng Chen. 2024a. Fedllm-bench: Realistic benchmarks for federated learning of large language models. *Advances in Neural Information Processing Systems*, 37:111106–111130.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. 2024b. 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.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D Lee, Wotao Yin, Mingyi Hong, and 1 others. 2024. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: a benchmark. In *Proceedings of the 41st International Conference on Machine Learning*, pages 59173–59190.

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

Haodong Zhao, Wei Du, Fangqi Li, Peixuan Li, and Gongshen Liu. 2023. Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Jujia Zhao, Wenjie Wang, Chen Xu, See Kiong Ng, and Tat-Seng Chua. 2025a. A federated framework for llm-based recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2852–2865.

Yanjun Zhao, Sizhe Dang, Haishan Ye, Guang Dai, Yi Qian, and Ivor Tsang. 2025b. Second-order fine-tuning without pain for LLMs: A hessian informed zeroth-order optimizer. In *International Conference on Learning Representations*.

Appendix

A Implementation details

A.1 Models, tasks, federated setups, training, and evaluations

We evaluate our proposed method with diverse natural language processing tasks. Specifically, we adopt OPT 1.3B (Zhang et al., 2022) and LLaMA 3.2-3B (Touvron et al., 2023) as the pretrained model for federated fine-tuning tasks. For evaluation tasks, we use the GLUE benchmark (Wang et al., 2018), which includes SST-2, SST-5, MNLI, RTE, and QNLI. Regarding the federated system setup, unless otherwise stated, we used $N = 20$ for fine-tuning OPT 1.3B, and used $N = 10$ for fine-tuning LLaMA 3.2-3B. We adopted full participation schemes, where all the clients participate in training in every communication round.

We use the prompt-based method for processing data instances during fine-tuning. We adopted prompt template from (Malladi et al., 2023; Gao et al., 2020), which is shown in Tab. 7.

For our framework, we use $\mu = 10^{-3}$, $q = 50$, $E = 5$ during ZO gradient estimation for our experiments. The simulated annealing for β_1 and β_2 follows (Jiang et al., 2024). The small positive number ε is set as 10^{-8}. For the learning rate η for each task, we perform hyperparameter tuning by searching from the set $\{1 \times 10^{-6}, 3 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-7}\}$. The reported performances are based on the results from the global test/evaluation. All experiments are conducted on NVIDIA V100 and A100 GPU Clusters.

To emulate heterogeneous data distributions, we used the symmetric Dirichlet distribution-based partition method with concentration parameter $\alpha = 0.5$, i.e. $Dir(0.5)$ (Guo et al., 2025a; Qin et al., 2024), where a smaller α indicates a higher degree of data heterogeneity and larger values of α tend to yield nearly identical (i.e., more homogeneous) distributions across clients. In particular, for each client, the distribution of each client is determined by a sampled stochastic vector, in which each entry indicates the class-wise ratio of data samples. The data partition scheme is implemented via sampling without replacement, which means there is no overlap across local datasets.

We compared our method with following state-of-the-art FL methods: ZO FL methods, including FedZO (Fang et al., 2022), FedKSeed (Qin et al., 2024), DeComFL (Li et al., 2025); and FFA LoRA (Sun et al., 2024a). We also implemented an FL version for the MeZO algorithm (Malladi et al., 2023), namely FedMeZO, in which local updates follow MeZO. For FO methods, we use the SGD optimizer. For ZO FL methods, we use the same number of perturbation vectors for fair comparison. Furthermore, we provided the performance of FedAvg (McMahan et al., Apr. 2017); this is provided for reference only, since it yields full model fine-tuning and transmission with significant computation and communication overhead. All baselines are implemented with the same FL system setup.

A.2 Random seed generations

As introduced in the main paper, we proposed a simple yet effective approach: Within each round, different clients use different random seed sequences without repeats for perturbation vector sampling. Now we elaborate on the implementation details.

Within communication round t, the server first generates a sequence of N random seeds without duplicates $\boldsymbol{s}_{\mathrm{G}}^t = [\boldsymbol{s}_{\mathrm{G}}^t[1], \boldsymbol{s}_{\mathrm{G}}^t[2], \ldots, \boldsymbol{s}_{\mathrm{G}}^t[N]]$, after which each random seed in $\boldsymbol{s}_{\mathrm{G}}^t$ is used to generate a list of qE distinct integers, for a total of NqE integers across N lists. Here, for any sequence $\boldsymbol{a}$, we let $\boldsymbol{a}[\ell]$ denote its ℓ-th entry. Any duplicate integer among these NqE integers will be replaced by sampling a new unique integer. Each list corresponds to a client. In particular, the integer list corresponding to client i is organized as a set $\boldsymbol{s}_i^t$ comprising E sequences, each of length q, i.e., $\boldsymbol{s}_i^t$ is denoted by

$$\boldsymbol{s}_i^t = \{(s_{\{i,k\},j}^t)_{j=1}^q\}_{k=1}^E, \tag{11}$$

where $\boldsymbol{s}_i^t$ is generated by the random seed $\boldsymbol{s}_{\mathrm{G}}^t[i]$. Upon finishing the generation of random integers, all the integers will be broadcast to the local clients, where each integer $s_{\{i,k\},j}^t$ is used as the random seed for sampling the j-th perturbation vector in the k-th step for client i in communication round t.

B Addition experimental results

B.1 Further sensitivity analysis

In this subsection, we provide additional sensitivity analysis on local ZO training steps E with fine-tuning OPT 1.3B over multiple datasets. As illustrated in Tab. 8, we can see that an increasing number of local steps can benefit to the overall performance, which is consistent to Theorem 1. While

Table 7: Prompt template and label words adopted in the experiments.

Datasets	**Prompt**	**Label words**
SST-2	$< S_1 >$ It was [MASK].	{great, terrible}
SST-5	$< S_1 >$ It was [MASK].	{great, good, okay, bad, terrible}
MNLI	$< S_1 >$? [MASK],$< S_2 >$	{Yes, Maybe, No}
RTE	$< S_1 >$? [MASK],$< S_2 >$	{Yes, No}
QNLI	$< S_1 >$? [MASK],$< S_2 >$	{Yes, No}

Table 8: Sensitivity analysis of our proposed framework with the task of fine-tuning OPT 1.3B.

Models	Setups	SST-2	SST-5	MNLI	RTE	QNLI
OPT 1.3B	$E = 1$	82.63	37.45	76.12	62.81	74.35
	$E = 5$	84.02	41.56	78.70	64.32	77.29
	$E = 10$	84.97	42.04	79.47	65.16	77.91
	$E = 20$	85.35	42.94	79.89	66.30	78.56

Table 9: Performance comparison of fine-tuning OPT 1.3B under diverse heterogeneous data settings.

Methods	SST-2			RTE		
	$\alpha = 10$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 10$	$\alpha = 0.5$	$\alpha = 0.1$
Ours (version from main paper)	84.21	84.02	83.87	64.82	64.32	64.15
Ours w/ Local adaptive gradients	84.29	84.23	84.10	64.85	64.52	64.39

we chose $E = 5$ in our main paper for all methods due to overhead. Increasing the number of local ZO steps brings additional local computation overhead as well as communication cost.

B.2 Variant of our proposed framework

Our proposed framework could be further enhanced by leveraging adaptive optimization for local updates. However, non-vanilla ZO local updates may lead to incompatible scalar-valued communication cost, since any adaptive gradient vector computed is not necessarily a linear combination of our perturbation vectors. In view of this, to leverage an adaptive gradient method for local updates while maintaining $\mathcal{O}(1)$ communication costs, we introduce a two-stage adaptive optimization-based global update, as summarized in Algorithm 2. Compared with the vanilla version, the variant algorithm effectively moves the computations for local gradient updates to the server. Specifically, the newly introduced client-wise adaptive update is performed between the gradient reconstructions and aggregation. Note that the aggregation will be performed via the usual adaptive global optimization with momentum. Detailed updates are given as follows:

1. Upon receiving the scalar-valued local ZO updates from client i, the server first reconstructs the local update, which is given by:

$$\widetilde{\nabla} f_i(\boldsymbol{w}_i^t) = \frac{1}{q} \sum_{k=1}^{E} \sum_{j=1}^{q} \Delta_{\{i,k\},j}^t \frac{\tilde{\boldsymbol{z}}_i(s_{\{i,k\},j}^t)}{\mu}. \tag{12}$$

2. After obtaining local gradients, the server performs adaptive gradient methods for each client instead of directly computing aggregated gradients. For convenience, the reconstructed local update $\widetilde{\nabla} f_i(\boldsymbol{w}_i^t)$ is simply denoted by $\boldsymbol{g}_i^t$. This client-wise adaptive update is computed as follows:

$$\boldsymbol{m}_i^t \leftarrow \beta_1 \boldsymbol{m}_i^{t-1} + (1-\beta_1)\boldsymbol{g}_i^t; \tag{13}$$

and for each ℓ-th entry ($1 \le \ell \le d$), we have

$$\boldsymbol{v}_i^t[\ell] \leftarrow \beta_2 \boldsymbol{v}_i^{t-1}[\ell] + (1-\beta_2)\boldsymbol{g}_i^t[\ell] \cdot \boldsymbol{g}_i^t[\ell]; \tag{14}$$

$$\boldsymbol{g}_i^t[\ell] \leftarrow \frac{\boldsymbol{m}_i^t[\ell]}{\sqrt{\boldsymbol{v}_i^t[\ell] + \varepsilon}}. \tag{15}$$

Algorithm 2 Efficient federated LLM fine-tuning with two-stage adaptive optimization

Inputs: N, T, $\boldsymbol{w}^1$ η, $\boldsymbol{m}^1$, $\boldsymbol{v}^1$, E
Outputs: Fine-tuned model w^{T+1}

for $t = 1$ **to** T **do**
  for each client $i = 1$ **to** N **in parallel do**
    // Zeroth-order model fine-tuning
    $\{(\Delta^t_{\{i,k\},j}, s^t_{\{i,k\},j})^q_{j=1}\}^E_{k=1} \leftarrow$ LocalZOUpdate($\boldsymbol{w}^t, \{s^t_{\{i,k\},j}\}$)
    Upload scalar pairs $\{(\Delta^t_{\{i,k\},j}, s^t_{\{i,k\},j})^q_{j=1}\}^E_{k=1}$ to server
    // Global ZO gradient processing
  Reconstruct ZO gradients via Eq. 12
  Update local gradients $\{\boldsymbol{g}^t_i\}^N_{i=1}$ via adaptive momentum method (Eq. 13 to Eq. 15)
  Compute gradients $\boldsymbol{g}^t$ via adaptive momentum method (Eq. 8 to Eq. 10 in main paper)
  // Global model update
  $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \eta\hat{\boldsymbol{g}}^t$
return $\boldsymbol{w}^T_{\mathrm{G}}$

function LocalZOUpdate($\boldsymbol{w}^t, \{s^t_{\{i,k\},j}\}$)
  Require: $\boldsymbol{w}^t$ is the latest global model. $\{s^t_{\{i,k\},j}\}$ is the seed sequence.
$w^t_{i,0} \leftarrow w^t$
for $j = 1$ **to** E **do**
  $\boldsymbol{w}^t_{i,k+1} \leftarrow \boldsymbol{w}^t_{i,k} - \eta\widehat{\nabla} f(\boldsymbol{w}^t_{i,k})$
  // Local ZO update with personalized perturbation
  Obtain ZO information $\{(\Delta^t_{\{i,k\},j}, s^t_{\{i,k\},j})^q_{j=1}\}^E_{k=1}$
return $\{(\Delta^t_{\{i,k\},j}, s^t_{\{i,k\},j})^q_{j=1}\}^E_{k=1}$

Here, $\boldsymbol{m}^t_i$ and $\boldsymbol{v}^t_i$ are local momentum-type vectors for client i.

3. After computing $\boldsymbol{g}^t_i$ for all i, the server shall perform aggregation to obtain the global gradient via $\boldsymbol{g}^t \leftarrow \frac{1}{N}\sum_{i=1}^N \boldsymbol{g}^t_i$. The subsequent model optimization is kept invariant.

Note also that this variant with local adaptive gradients further improves the local ZO fine-tuning performance (as illustrated in Tab. 9, especially in the context of highly heterogeneous data, while still keeping the local computation and communication invariant.

C Convergence analysis of the proposed framework

In this section, we present the proof of Theorem 1 in the main paper.

We first introduce the lemma that would be used in our proof.

Lemma 1 ((Li et al., 2025; Ghadimi and Lan, 2013)). *Given a smooth approximation of f_i as $f^\mu_i(\boldsymbol{w}) := \mathbb{E}[f_i(\boldsymbol{w} + \mu\boldsymbol{z})]$ with smoothing parameter μ and random perturbation vector $\boldsymbol{z}$, for any $\boldsymbol{w} \in \mathbb{R}^d$ we have*

$$\|\nabla f^\mu_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{w})\| \le \frac{1}{2}\mu L(d+3)^{\frac{3}{2}}, \tag{16}$$

$$\frac{1}{\mu^2}\mathbb{E}_{\boldsymbol{z}}\left[(f_i(\boldsymbol{x}+\mu\boldsymbol{z}) - f_i(\boldsymbol{x}))^2 \|\boldsymbol{z}\|^2\right] \le \tag{17}$$

$$\frac{\mu^2}{2}L^2(d+6)^3 + 2(d+4)\|\nabla f_i(\boldsymbol{x})\|^2. \tag{18}$$

Now we shall provide the proof of sketch for Theorem 1 given in our main paper.

Since $f_i(\cdot)$ is λ-smooth for $i = 1, \ldots, N$ and denote $\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta\hat{\boldsymbol{g}}^t$, we have

$$f(\boldsymbol{w}^{t+1}) \le f(\boldsymbol{w}^t) - \eta\left\langle \nabla f(\boldsymbol{w}^t), \hat{\boldsymbol{g}}^t\right\rangle + \frac{L\eta^2}{2}\|\hat{\boldsymbol{g}}^t\|^2. \tag{19}$$

We first bound $\hat{\boldsymbol{g}}^t$ with

$$\|\hat{\boldsymbol{g}}^t\|^2 = \sum_{\ell=1}^d \frac{(\boldsymbol{m}^t[\ell])^2}{\boldsymbol{v}^t[\ell] + \varepsilon} \le \frac{\|\boldsymbol{m}^t\|^2}{\varepsilon} \le \frac{G^2}{\varepsilon} \tag{20}$$

By introducing the smoothed gradient $\nabla f^{\mu} = \frac{1}{N}\sum_i \nabla f_i^{\mu}$, we then have

$$\boldsymbol{g}^t - \nabla f(\boldsymbol{w}^t) = \underbrace{\boldsymbol{g}^t - \nabla f^{\mu}(\boldsymbol{w}^t)}_{R_1} + \underbrace{\nabla f^{\mu}(\boldsymbol{w}^t) - \nabla f(\boldsymbol{w}^t)}_{R_2}. \quad (21)$$

For term R_1, we can further write it as

$$\boldsymbol{g}^t - \nabla f^{\mu}(\boldsymbol{w}^t) = \frac{1}{N}\sum_{i=1}^{N}\left[\widetilde{\nabla} f_i(\boldsymbol{w}_i^t) - \nabla f_i^{\mu}(\boldsymbol{w}^t)\right] + \frac{1}{N}\sum_{i=1}^{N}\left[\nabla f_i^{\mu}(\boldsymbol{w}^t) - \nabla f^{\mu}(\boldsymbol{w}^t)\right]. \quad (22)$$

Applying Young's inequality, we have

$$\mathbb{E}\left[\|\boldsymbol{g}^t - \nabla f^{\mu}(\boldsymbol{w}^t)\|^2\right] \leq 2\,\mathbb{E}\left[\left\|\frac{1}{N}\sum_i n_i\right\|^2\right] + 2\,\mathbb{E}\left[\left\|\frac{1}{N}\sum_i d_i\right\|^2\right], \quad (23)$$

in which $n_i = \frac{1}{E}\sum_{k=0}^{E-1}\left[\widetilde{\nabla} f_i(\boldsymbol{w}_{i,k}^t) - \nabla f_i^{\mu}(\boldsymbol{w}_{i,k}^t)\right]$ denotes the noise and $d_i = \frac{1}{E}\sum_{k=0}^{E-1}\left[\nabla f_i^{\mu}(\boldsymbol{w}_{i,k}^t) - \nabla f_i^{\mu}(\boldsymbol{w}^t)\right]$ denotes the local drift.

After applying Assumption 2 and Lemma 1, we can bound the first term in (23):

$$\mathbb{E}\left[\left\|\frac{1}{N}\sum_i n_i\right\|^2\right] = \frac{1}{N^2}\sum_i \mathbb{E}\|n_i\|^2 \leq \frac{c_d B_q}{NEq}, \quad (24)$$

in which $B_q := \frac{\mu^2 L^2 (d+6)^3}{2} + 2(d+4)G^2 + \sigma_b^2$ and c_d is a constant.

Similarly, we further bound the second term in (23) as:

$$\left\|\frac{1}{N}\sum_i d_i\right\|^2 \leq \frac{1}{N}\sum_i \|d_i\|^2 \leq \frac{L^2\eta^2 G^2 E^2}{3}. \quad (25)$$

Combining the above two terms, we have

$$\mathbb{E}\left[\|\boldsymbol{g}^t - \nabla f^{\mu}(\boldsymbol{w}^t)\|^2\right] \leq \frac{2c_d B_q}{NEq} + \frac{2L^2\eta^2 G^2 E^2}{3}. \quad (26)$$

Furthermore, R_2 could be bound as

$$\|\nabla f^{\mu}(\boldsymbol{w}^t) - \nabla f(\boldsymbol{w}^t)\|^2 \leq \frac{1}{N}\sum_{i=1}^{N}\|\nabla f_i^{\mu} - \nabla f_i\|^2 \leq \frac{\mu^2 L^2 (d+3)^3}{4}. \quad (27)$$

Combining them, we have

$$\mathbb{E}\left[\|\boldsymbol{g}^t - \nabla f(\boldsymbol{w}^t)\|^2\right] \leq \frac{4c_d B_q}{NEq} + \frac{4L^2\eta^2 G^2 E^2}{3} + \frac{\mu^2 L^2 (d+3)^3}{2} = \Delta_0 \quad (28)$$

We now bound the inner product term $\langle \nabla f(\boldsymbol{w}^t), \hat{\boldsymbol{g}}^t \rangle$:

$$\langle \nabla f, \frac{\boldsymbol{m}^t}{\sqrt{\boldsymbol{v}^t + \varepsilon}} \rangle = \underbrace{\sum_{\ell=1}^{d} \frac{(\nabla f[\ell])^2}{\sqrt{\boldsymbol{v}^t[\ell] + \varepsilon}}}_{R_3} + \underbrace{\sum_{\ell=1}^{d} \frac{\nabla f[\ell]\,(\boldsymbol{m}^t[\ell] - \nabla f[\ell])}{\sqrt{\boldsymbol{v}^t[\ell] + \varepsilon}}}_{R_4}. \quad (29)$$

We further have $R_3 \geq \|\nabla f\|^2 / \sqrt{G^2 + \varepsilon}$ and

$$R_4 \geq -\frac{|\nabla f|^2}{2\sqrt{G^2 + \varepsilon}} - \frac{\sqrt{G^2 + \varepsilon}}{2\varepsilon}\|\boldsymbol{m}^t - \nabla f\|^2. \quad (30)$$

Also, we have

$$\begin{aligned} &\|\boldsymbol{m}^t - \nabla f(\boldsymbol{w}^t)\|^2 \\ \leq &2\|\boldsymbol{m}^t - \boldsymbol{g}^t\|^2 + 2\|\boldsymbol{g}^t - \nabla f(\boldsymbol{w}^t)\|^2 \\ \leq &8\beta_1^2 G^2 + 2\|\boldsymbol{g}^t - \nabla f(\boldsymbol{w}^t)\|^2, \end{aligned} \quad (31)$$

Combining R_3, R_4, and (28), we have

$$\mathbb{E}\left[\langle \nabla f(\boldsymbol{w}^t), \hat{\boldsymbol{g}}^t \rangle\right] \geq \frac{1}{2\sqrt{G^2 + \varepsilon}}\mathbb{E}\|\nabla f(\boldsymbol{w}^t)\|^2 - \frac{\sqrt{G^2 + \varepsilon}}{2\varepsilon}\left(8\beta_1^2 G^2 + 2\Delta_0\right). \quad (32)$$

Substituting two term into the first expansion (19), we have

$$\mathbb{E}[f(\boldsymbol{w}^{t+1})] \leq \mathbb{E}[f(\boldsymbol{w}^t)] - \frac{\eta}{2\sqrt{G^2 + \varepsilon}}\mathbb{E}\|\nabla f(\boldsymbol{w}^t)\|^2 + \frac{\eta\sqrt{G^2 + \varepsilon}}{2\varepsilon}(8\beta_1^2 G^2 + 2\Delta_0) + \frac{L\eta^2 G^2}{2\varepsilon}. \quad (33)$$

Summing from $T = 1$ to T and using the telescoping sum, we can obtain the bound.