
Towards Straggler-Resilient Split Federated Learning: An Unbalanced Update Approach

Dandan Liang

Rochester Institute of Technology
Rochester, New York
dl15974@rit.edu

Jianing Zhang

Purdue University
West Lafayette, Indiana
zhan4670@purdue.edu

Evan Chen

Purdue University
West Lafayette, Indiana
chen4388@purdue.edu

Zhe Li

Rochester Institute of Technology
Rochester, New York
z14063@rit.edu

Rui Li

Rochester Institute of Technology
Rochester, New York
rxlics@rit.edu

Haibo Yang

Rochester Institute of Technology
Rochester, New York
hbycis@rit.edu

Abstract

Split Federated Learning (SFL) enables scalable training on edge devices by combining the parallelism of Federated Learning (FL) with the computational offloading of Split Learning (SL). Despite its great success, SFL suffers significantly from the well-known straggler issue in distributed learning systems. This problem is exacerbated by the dependency between Split Server and clients: the Split Server side model update relies on receiving activations from clients. Such synchronization requirement introduces significant time latency, making straggler a critical bottleneck to the scalability and efficiency of the system. To mitigate this problem, we propose MU-SplitFed, a straggler-resilient SFL algorithm in zeroth-order optimization that decouples training progress from straggler delays via a simple yet effective unbalanced update mechanism. By enabling the server to perform τ local updates per client round, MU-SplitFed achieves a convergence rate of $\mathcal{O}(\sqrt{d}/(\tau T))$ for non-convex objectives, demonstrating a linear speedup of τ in communication rounds. Experiments demonstrate that MU-SplitFed consistently outperforms baseline methods with the presence of stragglers and effectively mitigates their impact through adaptive tuning of τ . The code for this project is available at <https://github.com/Johnny-Zip/MU-SplitFed>.

1 Introduction

Split Federated Learning (SFL) [1–3] integrates the strengths of Federated Learning (FL) [4] and Split Learning (SL) [5, 6], enabling efficient training on resource-constrained devices. FL offers parallel client updates but imposes heavy computation on edge devices [7], while SL reduces client load by offloading computation to the server but suffers from high latency due to its sequential nature. SFL balances these trade-offs, making it a promising framework for scalable training, especially as model sizes grow. However, the relay-based training mechanism in SFL introduces synchronization bottlenecks due to stragglers: clients with the slowest computation or communication speeds delay the overall process [8, 9]. Both global aggregation and client-side updates must wait for the slowest participant, limiting scalability [10]. This issue is a well-known bottleneck in distributed learning that

can severely degrade training efficiency [11–14]. This issue is further exacerbated by the increasing size of ML models and the limited computational capacity of edge devices [15].

To address this issue, existing works draw inspiration from straggler mitigation strategies in FL. Adaptive splitting techniques [9, 16] dynamically adjust the client-side cut layer based on network conditions to enforce synchronization. However, this strategy requires the model architecture to expose layers with varying activation dimensions. In modern transformer-based models, where activation sizes are nearly uniform across layers, such flexibility is absent, and shifting the cut therefore provides little benefit: the amount of data transmitted remains essentially constant, leading to persistent communication delays regardless of the split location. Another approach enables asynchronous updates by allowing the server to proceed with stale information [8]. While this reduces idle time, it exacerbates client drift under high data heterogeneity, harming model performance. Although these methods focus on reducing the straggler-induced latency, they often overlook a more dominant factor contributing to training overhead: the total global communication round. As a result, we investigate the following question: *Can we **efficiently** alleviate the impact of stragglers in SFL by strategically **reducing** communication round?*

We provide an affirmative answer in this paper, aiming to accelerate convergence under practical system heterogeneity and thereby reduce training time overhead. We propose MU-SplitFed, a SFL framework that leverages unbalanced server-client updates to improve training efficiency by controlling communication frequency. Our approach exploits the computational advantage of powerful servers: instead of idly waiting for slow edge devices, the Split Server performs τ optimization steps for each client-server communication round, effectively accelerating the training process. To further ease memory and computation burdens on edge devices, we incorporate Zeroth-Order (ZO) optimization on the client side, enabling training without backpropagation [17, 18]. Beyond empirical performance, we provide a rigorous theoretical analysis showing that our method achieves *linear speedup with respect to the server iteration τ* , without relying on strong assumptions. As a result, the total training time is no longer affected by the speed of the slowest client. Our analysis also rigorously accounts for the variance introduced by ZO methods. Due to model splitting, obtaining tight convergence bounds for SFL is more challenging than for standard FL: existing theoretical results in parallel SFL [9] use stronger assumptions (e.g., bounded gradients) while failing to capture the acceleration from clients or local updates. In contrast, our theoretical results not only reflect the acceleration from τ but also account for other factors such as the number of clients.

We summarize our main contributions as follows:

- **A novel SFL framework:** We propose MU-SplitFed, a straggler-resilient SFL framework that effectively reduces the communication round by leveraging unbalanced server-client updates. While other SFL methods suffer from server idleness due to stragglers, MU-SplitFed enables the server to perform τ local updates during each client-server communication round. This effectively utilizes server-side computation and decouples total training time from the slowest client. By incorporating ZO optimization, our method further reduces resource usage on low-capacity edge devices (Sec. 3).
- **Theoretical convergence with linear speedup:** We provide a rigorous convergence analysis of MU-SplitFed. The convergence rate is $\mathcal{O}(\sqrt{d}/(\tau T))$ for non-convex setting with the standard assumptions, showing the linear speedup w.r.t the server-side update τ . Furthermore, our theory supports that the reduction in the communication round allows the total training time to become independent of the straggler’s speed, directly addressing a major bottleneck in SFL (Sec. 4.2).
- **Insights into model partitioning and update alignment:** We uncover a critical connection between the model splitting strategy and the unbalanced update ratio. Both our theoretical and empirical results demonstrate that aligning the server-side model depth with the value of τ is essential for optimal convergence. A larger τ would benefit from more layers on the server side, thus accelerating convergence through more effective server-side computation (Sec. 4.1).
- **Empirical validation:** We validate the effectiveness of MU-SplitFed through experiments on benchmark datasets. Beyond its advantage in reducing communication round, our method consistently outperforms baselines under high client heterogeneity, highlighting its practical feasibility for straggler mitigation in SFL (Sec. 5).

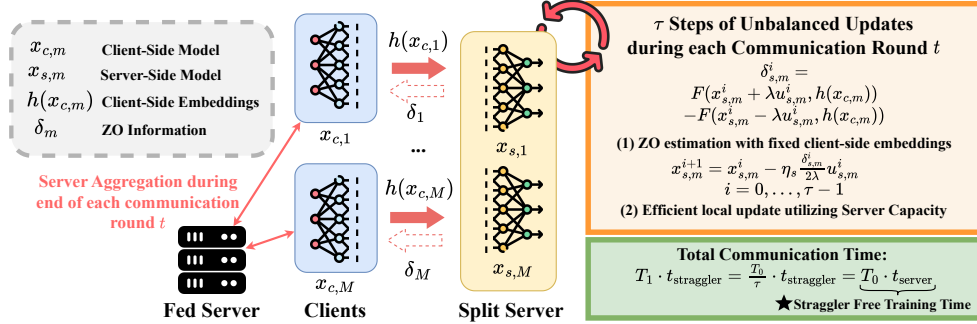


Figure 1: Overview of MU-SplitFed. The global model x is split at the cutting layers into two parts: client-side model x_c and server-side model x_s . Each client m trains its local copy $x_{c,m}$ while the Split Server performs τ local updates on $x_{s,m}$ using the latest embedding, without waiting for the client to finish. At the end of each global round, the Fed Server aggregates all client-side models, and the Split Server averages all the server-side models to form the updated global model.

2 Background and Motivation

SFL Setup. We consider the parallel SFL framework [1] which combines the model-splitting strategy of SL [5] with the parallel client updates of FL [4]. In SFL, a neural network is partitioned at layer L_c , assigning the first L_c layers to the M clients as “client-side model”, parameterized by $\{x_{c,1}, \dots, x_{c,M}\}$, and the remaining layers to the *Split Server*, which maintains M corresponding “server-side model” $\{x_{s,1}, \dots, x_{s,M}\}$. The combined parameters for client m are denoted as $x_m = \{x_{c,m}, x_{s,m}\}$. Client m computes the embedding $h_m = h(x_{c,m}; \xi_m)$ at the cut layer and sends it to the *Split Server*, which holds the label y_m and computes the loss:

$$F(x_m; \xi_m) = F(x_{s,m}, h(x_{c,m}; \xi_m); y_m), \quad (1)$$

where $\xi_m \sim \mathcal{D}_m$ is the data sample as client input. The server computes a gradient estimate and returns it to the client, which uses it to update both client-side and server-side parameters. The M client-server pairs collaboratively train a global model. After each round of local training, the client-side models are aggregated by the *Fed Server*, while the server-side models are aggregated by the *Split Server*. The overall objective of the SFL framework can be formulated as:

$$\min_x f(x) := \sum_{m=1}^M w_m f_m(x), \quad (2)$$

where $f_m(x) = \frac{1}{|\mathcal{D}_m|} \sum_{\xi \in \mathcal{D}_m} F(x; \xi)$ is the local loss function, and w_m is the weight of client m , with $w_m \in [0, 1]$ satisfying $\sum_{m=1}^M w_m = 1$.

ZO Optimization. Zeroth-Order Optimization (ZOO) is a gradient-free method, offering an alternative solution for scenarios where explicit gradient computation is impractical, expensive, or unreliable [19–21]. ZOO has shown significant advantages in memory saving because it requires only forward passes [17, 18]. Since our goal is to improve training efficiency for edge devices with limited memory resources, we adopt ZOO to reduce even more memory consumption for our resource-constrained devices. In specific, we adopt Simultaneous Perturbation Stochastic Approximation (SPSA) [22] as our ZO gradient estimator. Let u be uniformly sampled from the Euclidean sphere $\sqrt{d}\mathbb{S}^{d-1}$, for any function $f(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ and $\lambda > 0$, we define its ZO gradient estimator as:

$$g(x) = \frac{f(x+\lambda u) - f(x-\lambda u)}{2\lambda} u \quad (3)$$

Challenges in Mitigating Stragglers in SFL. The straggler problem is a persistent bottleneck in distributed learning systems, where synchronous training requires coordinated updates across multiple agents [9, 10, 23]. In SFL, this issue is further exacerbated by the interdependence between clients and server. There are two factors that contribute to this severity: 1) the server must wait for all clients to transmit embeddings or gradients before continuing, making the system highly sensitive to the slowest participant; 2) the model is split across client and server, requiring frequent communication during both forward and backward passes. This tight coupling amplifies the impact of stragglers compared to traditional FL, where delays are typically limited to full model updates.

In FL, asynchronous updates have been proposed to mitigate such issues by decoupling client updates from global synchronization [23–27]. However, these approaches are insufficient for SFL,

as they only address global aggregation. In SFL, the straggler problem also arises from split-layer communication, a fundamental difference that makes asynchronous techniques in FL less effective when directly applied to SFL. Recent efforts in SFL have explored adaptive model partitioning to balance computation and communication delays [8, 9, 16]. These methods are constrained by the network architecture and fail to address the core issue: the high communication frequency between clients and the server. As a result, none of the existing straggler solutions explicitly aim to reduce the number of communication between client and Split Server, which is the key problem to SFL’s straggler-induced inefficiency. *These limitations point to the need for a new framework that explicitly exploits SFL’s structural properties to reduce communication frequency, thereby mitigating stragglers without sacrificing model performance.*

3 Methodology

Building upon the aforementioned challenges, we propose MU-SplitFed to mitigate the straggler issue by jointly addressing memory inefficiency, computation imbalance, and communication overhead. By combining unbalanced update scheduling and zeroth-order optimization, our algorithm achieves robust and scalable performance tailored for resource-constrained edge devices.

Algorithm 1: MU-SplitFed

Input: Unbalanced update steps τ , global communication rounds T , local learning rate on server side η_s , learning rate on client side η_c
Output: Global model $x^T = \{x_c^T, x_s^T\}$

- 1 **each global round** $t = 0, \dots, T - 1$ **do**
- 2 **each client** $m \in \{1, 2, \dots, M\}$ **in parallel do**
- 3 Pull global model for initialization: $x_{c,m}^t \leftarrow x_c^t, x_{s,m}^{t,0} \leftarrow x_s^t;$
- 4 /* Phase 1: Unbalanced Update on Split Server */
- 5 **each client** $m \in \{1, 2, \dots, M\}$ **in parallel do**
- 6 Send embeddings $h_{c,m}^{t+}, h_{c,m}^{t-}$ to the Split Server;
- 7 **each local iteration** $i = 0, \dots, \tau - 1$ **do**
- 8 Compute zeroth-order gradient $g_{s,m}^{t,i}$ according to (5);
- 9 Update Split Server model: $x_{s,m}^{t,i+1} \leftarrow x_{s,m}^{t,i} - \eta_s G_{s,m}^{t,i};$
- 10 Compute zeroth-order info $\delta_{c,m}^t$ according to (6) and send it back to the client;
- 11 Update client model: $x_{c,m}^{t+1} \leftarrow x_{c,m}^t - \eta_c G_{c,m}^t(\mathbf{x}^t);$
- 12 /* Phase 2: Model Aggregation on Fed Server */
- 13 Fed Server and Split Server updates according to (7), Fed Server broadcasts x_c^{t+1} to all clients.

Training Procedures. MU-SplitFed integrates an unbalanced update strategy and ZO optimization into the SFL framework. The overall training process consists of two main phases: 1) *Unbalanced ZO updates between clients and Split Server:* A subset of clients communicates with their corresponding server-side models on the Split Server and performs local training using ZO optimization in an unbalanced update manner. 2) *Federated Aggregation across M models:* The Fed Server collects the updated model weights x_m for $m \in [M]$ and applies the FedAvg strategy to compute a new global model. We detail both phases below and provide the full procedure in Algorithm 1.

Client Model Perturbation and Forwarding. At global round t , each activated client m samples a data point $\xi_m^t \in \mathcal{D}_m$. To perform ZO updates, the client perturbs its model parameters and computes the corresponding embeddings multiple times. First, the client computes the unperturbed embedding $h_m^t = h(x_{c,m}^t; \xi_m^t)$, and the perturbed embeddings:

$$h_m^{t+} = h(x_{c,m}^t + \lambda u_{c,m}^t; \xi_m^t), \quad \text{and} \quad h_m^{t-} = h(x_{c,m}^t - \lambda u_{c,m}^t; \xi_m^t), \quad (4)$$

where $u_{c,m}^t$ is the perturbation direction sampled according to Equation (3), λ is a smooth parameter, $x_{c,m}^t$ is the client-side model at round t . We define $H_m^t = \{h_m^t, h_m^{t+}, h_m^{t-}\}$. The client then transmits H_m^t to the server for computing the ZO gradient required for model updates.

Unbalanced Split Server Update. The transmission of embeddings follows an on-the-fly manner: each embedding is sent immediately after it is computed. Unlike the client, which requires feedback from the server to proceed updates, the Split Server can compute ZO gradients independently. To fully utilize the server’s computational capacity, we introduce an unbalanced update mechanism, allowing the server to perform multiple updates using the unperturbed embedding h_m^t . Specifically,

instead of remaining idle, the server initiates multiple local updates using h_m^t , while waiting for the full set of perturbed embeddings h_m^{t+} and h_m^{t-} . We denote $i = 0, 1, \dots, \tau - 1$ as the server update round. At global round t and server round i , the server perturbs its model parameters and computes the corresponding ZO gradient differences:¹

$$\delta_{s,m}^{t,i} = F(x_{s,m}^{t,i} + \lambda u_{s,m}^{t,i}, h_m^t) - F(x_{s,m}^{t,i} - \lambda u_{s,m}^{t,i}, h_m^t), \quad (5)$$

where $u_{s,m}^{t,i}$ is sampled according to (3), and $x_{s,m}^{t,i}$ denotes the server-side model parameters for client m at global round t and server update step i . The corresponding ZO gradient estimator is computed as: $G_{s,m}^{t,i} = \frac{\delta_{s,m}^{t,i}}{2\lambda} u_{s,m}^{t,i}$, where $\delta_{s,m}^{t,i}$ denotes the loss difference obtained from the perturbed embeddings. The server-side model is updated iteratively over τ local steps using the ZO oracle: $x_{s,m}^{t,i+1} = x_{s,m}^{t,i} - \eta_s^t G_{s,m}^{t,i}$, $i \in [0, \tau)$.

Zeroth-order Back Propagation and Client Update. After completing server-side local updates, it then computes the ZO loss differences required for client-side model updates:

$$\delta_{c,m}^t = F(x_{s,m}^{t,\tau}, h_m^{t+}) - F(x_{s,m}^{t,\tau}, h_m^{t-}), \quad (6)$$

where each $\delta_{c,m}^t$ is a scalar and incurs minimal communication overhead. These ZO differences are sent back to the client. Clients compute their ZO estimates as $G_{c,m}^t = \frac{\delta_{c,m}^t}{2\lambda} u_{c,m}^t$, and update their models via $x_{c,m}^{t+1} = x_{c,m}^t - \eta_c^t G_{c,m}^t$.

Global Aggregation. At the end of the global communication round t , once all activated local models $x_m = \{x_{c,m}, x_{s,m}\}$ has completed their update, the Fed Server collects the updated parameters $x_{c,m}$ and performs model aggregation, while the Split Server also locally aggregates $x_{s,m}$ and performs an update on x_s :

$$x_c^{t+1} = x_c^t - \eta_g \sum_m w_m (x_{c,m}^{t+1} - x_{c,m}^t), \quad \text{and} \quad x_s^{t+1} = x_s^t - \eta_g \sum_m w_m (x_{s,m}^{t,\tau} - x_{s,m}^t), \quad (7)$$

where w_m denotes the aggregation weight for client m , in our algorithm we choose to set $w_m = \frac{1}{M}$. η_g is the learning rate for global update. Then, the Fed Server broadcasts x_c^{t+1} to all clients.

4 Convergence Analysis

In this section, we present a rigorous convergence analysis of MU-SplitFed. Specifically, we want to quantify the effect of our unbalanced update mechanism on convergence. However, in FL, this effect may be intertwined with other factors such as data and system heterogeneity. To isolate the influence of the unbalanced updates, we first analyze the single-client setting, which simplifies to a standard SL framework (Sec. 4.1). Then, we propose our general result under SFL settings (Sec. 4.2). The complete proofs are deferred to Appendix C and D. Here, we first make some standard assumptions that will facilitate our analysis.²

Assumption 4.1 (*L-Smooth*). The loss function f is bounded from below, and is L -smooth, i.e. $\forall x, y, \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.

Assumption 4.2 (*Bounded Variance*). The variance of the stochastic gradient w.r.t. the client and the server is upper-bounded by σ_c^2 and σ_s^2 . Specifically, for $\forall \xi \in \mathcal{D}_m, \|\nabla_{x_c} f(x; \xi) - \nabla_{x_c} f(x)\|^2 \leq \sigma_c^2$ and $\|\nabla_{x_s} f(x; \xi) - \nabla_{x_s} f(x)\|^2 \leq \sigma_s^2$.

4.1 Convergence Analysis for MU-Split

To analyze the impact of multiple server updates alone, we consider the special case where $M = 1$, denoted as MU-Split, which reduces to the SL setting. The convergence of MU-Split is established in the following theorem:

Theorem 4.1. *Under Assumption 4.1 and 4.2, and let the server iteration number be τ . If the learning rates on client and server satisfy $\eta_c/\tau = \eta_s = \eta \leq \min\{\frac{1}{64L(\tau+2d_s)}, \frac{1}{16L\tau d_c}\}$, the sequence of iterates generated by our MU-Split satisfies:*

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^T \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] &\leq \frac{4\mathcal{F}}{\eta\tau T} + 16\eta L(\eta\tau L + 1)d_s\sigma_s^2 + 8\eta\tau L d_c\sigma_c^2 \\ &\quad + 4L^2(\eta^2\tau^2L^2 + 1/4)\lambda^2 d_s^3 + L^2\lambda^2 d_c^3, \end{aligned} \quad (8)$$

¹Here we slightly abuse the notation and denote $F(x_{s,m}^{t,i}) = F(x_{s,m}^{t,i}, h(x_{c,m}^t, \xi_m^t); y_m^t)$, where y_m^t is the label corresponding to data ξ_m^t .

²The assumptions adopted in our analysis are standard and consistent with those commonly used in the distributed optimization literature [28–30]. We focus on the non-convex setting.

where $\mathcal{F} = \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)]$; d_c and d_s represent the dimensions of the parameters on the client and server side, respectively; $d = d_c + d_s$ is the total number of parameters. λ is the smooth parameters for ZO Oracle defined in (3), and σ_c^2, σ_s^2 are the upper bound of the gradient variance on client and server, respectively. $\eta = \eta_c/\tau = \eta_s$ is the unified learning rate.

To establish the theorem, the learning rate on server needs to shrink linearly with multiple update steps τ , i.e. $\eta_c/\tau = \eta_s$. This requirement stems from the need to balance client and server progress: since the server performs τ updates for each client update, a proportionally smaller server learning rate ensures synchronized convergence. The convergence bound in equation (8) contains five distinct terms, each capturing different aspects of the algorithm's behavior.

The first term, $\frac{4\mathcal{F}}{\eta\tau T}$, represents the optimization error and decays as either the total number of communication rounds T or the server-side update frequency τ increases. This rate matches the same rate as typical ZO-SGD methods when $\tau = 1$, which generalizes the classical convergence rate without unbalanced update. It also highlights the benefit of unbalanced updates: increasing the number of server iterations per round leads to a faster reduction of this term. *This demonstrates the improved convergence behavior enabled by unbalanced server updates.*

The second and third terms quantify the error introduced by the variance of the stochastic gradient estimates on the server and client, respectively. Notably, those two terms scales up with the parameter τ . This means that a larger τ exacerbates the stochastic error, thus leading to high variance in the estimated gradient that hinders convergence performance. To keep these terms small, an inverse relationship between the Split training learning rate and server-side local steps should be satisfied, i.e., $\eta_s = \eta = \mathcal{O}(1/\sqrt{\tau})$. Specifically, note that both the server-side and client-side variances are linearly amplified by τ . This requires a sufficiently small η to offset the variance between two successive communication rounds to make the those error term in small. The intuitive explanation behind this is that when the server applies multiple consecutive updates using outdated client information, it introduces client drift and allows stochastic errors to accumulate progressively. Consequently, smaller step sizes are required to balance the impact of these accumulated error terms.

The last two terms, $4L^2(\eta^2\tau^2L^2 + 1/4)\lambda^2d_s^3 + L^2\lambda^2d_c^3$ capture errors introduced by the zeroth-order gradient estimation. These terms are independent of the learning rate choice and decrease as the smoothing parameter λ decreases, indicating that more accurate ZO gradient estimation improves overall convergence.

We can further derive a convergence rate for all terms if certain conditions are met.

Corollary 4.2. *Based on Theorem 4.1, let the model split satisfies $d_c = \sqrt{d/\tau}, d_s = d - \sqrt{d/\tau}$; let $\tau \leq d$, the smoothing parameter satisfies $\lambda^2 \leq \frac{1}{\sqrt{\tau T}d^{5/2}L}$, and choose the unified learning rate as $\eta \leq \min\{\frac{1}{64L(\tau+2d_s)}, \frac{1}{16L\tau d_c}, \frac{1}{\sqrt{d\tau T}}\}$. Then we have the following convergence rate:*

$$\frac{1}{T} \sum_{t=0}^T \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] \leq \frac{4\sqrt{d}\mathcal{F}}{\sqrt{\tau T}} + \frac{48L\sqrt{d}\sigma_s^2}{\sqrt{\tau T}} + \frac{9\sqrt{d}}{\sqrt{\tau T}} + \frac{8L\sigma_c^2}{\sqrt{T}} \quad (9)$$

Discussion. All dominant terms in equation (9) converge at the rate of $\mathcal{O}(\sqrt{d/\tau T})$, when we choose $d_c = \sqrt{d/\tau}$ and $d_s = d - \sqrt{d/\tau}$, where $d = d_c + d_s$ is the total number of parameters. This rate highlights a linear speedup in term of τ .³ *The linear speedup is achieved when the client-side parameter dimension d_c scales as $\mathcal{O}(d/\sqrt{\tau})$.* This has direct implications for network architecture design in split learning systems. In particular, when the server has higher computational capacity, it is beneficial to allocate fewer parameters to the client side, thereby placing the split closer to the input layer. That's being said, ZOO provides a natural mechanism for controlling stochastic variance through the cutting layer strategy. By connecting the cutting layer choice with multiple server updates steps τ , the variance impact on the client side is effectively reduced. This variance reduction occurs because fewer layers are processed on the client side, which inherently limits the accumulation of gradient estimation errors. This theoretical finding aligns with our empirical observations in the ablation study presented in Section 5.

³To attain ε accuracy for an algorithm, it needs $\mathcal{O}(\frac{1}{\varepsilon^2})$ communication rounds with a convergence rate $\mathcal{O}(\frac{1}{\sqrt{T}})$, while needing $\mathcal{O}(\frac{1}{\tau\varepsilon^2})$ rounds if the convergence rate is $\mathcal{O}(\frac{1}{\sqrt{\tau T}})$. In this sense, one achieves a linear speedup with respect to τ .

4.2 Convergence Analysis for MU-SplitFed

We further derive the following convergence result for MU-SplitFed under SFL with M clients. For the convergence analysis of MU-SplitFed under SFL, we further assume that the above two assumptions apply to f_m for $\forall m \in [M]$. To quantify the data heterogeneity across clients, we make the following assumption on data distribution:

Assumption 4.3 (Bounded Heterogeneity). For $\forall m \in [M]$, the global variability of the local gradient is upper bounded: $\|\nabla f_m(x) - \nabla f(x)\|^2 \leq \epsilon^2$.

Theorem 4.3. Under Assumption 4.1 to 4.3, consider a SFL framework with M clients, and let the server iteration number be τ . If the learning rates on client and server satisfy $\eta_c/\tau = \eta_s = \eta \leq \min\{\frac{1}{120L\tau(1+2d_s/\tau)}, \frac{M}{12\tau Ld_c}\}$, the sequence of iterates generated by MU-Split satisfies:

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^T \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] &\leq \frac{4\mathcal{F}}{T\eta_g\eta\tau} + 16\eta(2\eta\tau L + \eta_g/M)Ld_s\sigma_s^2 + \frac{4\eta_g\eta\tau Ld_c\sigma_c^2}{M} \\ &\quad + 24\eta(4\eta\tau L + \eta_g/M)L(\tau + 2d_s)\epsilon^2 + \frac{12\eta_g\eta\tau Ld_c\epsilon^2}{M} \\ &\quad + (1/\tau + 8\eta^2\tau L^2 + 2\eta_g\eta/M)\tau L^2\lambda^2 d_s^3 + L^2\lambda^2 d_c^3 \end{aligned} \quad (10)$$

where $\mathcal{F} = \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)]$; d_c and d_s represent the dimensions of the parameters on the client-side and server-side, respectively; λ is the smooth parameters for ZO Oracle defined in (3), and σ_c^2, σ_s^2 are the upper bound of the gradient variance on client and server. Additionally, η_g is the global learning rate for model aggregation, and ϵ^2 quantifies data heterogeneity.

The first term and the last two terms are similar to MU-Split, which are attributed to model initialization and ZO optimization. Compared to traditional SFL, the presence of server iteration τ is again observed on the denominator, which corresponds to our observation in MU-Split: convergence is accelerated by multiple server updates. The second and third terms correspond to the variance of the stochastic gradient estimator on the server and client, respectively. Again, both terms scales with the increase of τ , which is consistent with MU-Split. In contrast to the analysis in MU-Split, the fourth and fifth terms are newly introduced to account for data heterogeneity, and they are also observed in other Federated Learning literature. Notably, those two terms scales with the parameter τ . This means that a larger τ exacerbates the heterogeneity error thus leading to increases client drift consequently. So, similar to SL, to offset the variance introduced by data heterogeneity and stochastic gradient estimation, a sufficiently small η should be selected and decay linearly with τ .

Corollary 4.4. Based on Theorem 4.3, if we further ensure that the neural network is cut such that $d_c = \sqrt{d/\tau}, d_s = d - \sqrt{d/\tau}$; let $\tau \leq d$, let the smoothing parameter $\lambda^2 \leq \frac{1}{\sqrt{\tau T d^{5/2} L}}$, and choose learning rate as $\eta \leq \min\{\frac{1}{120L\tau(1+2d_s/\tau)}, \frac{M}{12\tau Ld_c}, \frac{1}{L\tau\sqrt{dT}}\}$, $\eta_g = \sqrt{\tau M}$. Define $\mathcal{F} = \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)]$, and we have the following bound:

$$\frac{1}{T} \sum_{t=0}^T \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] \leq \frac{4L\sqrt{d}\mathcal{F}}{\sqrt{\tau TM}} + \frac{8\sqrt{d}(3\epsilon^2 + 2\sigma_s^2)}{\sqrt{\tau TM}} + \frac{32\sqrt{d}(3\epsilon^2 + \sigma_s^2)}{\tau T} + \frac{12\epsilon^2 + 4\sigma_c^2}{\sqrt{\tau TM}} + \frac{6\sqrt{d}}{\tau T} \quad (11)$$

Discussion. The first and second term converge at the rate $\mathcal{O}(\sqrt{d}/(\tau TM))$. Compared with MU-Split, the involvement of multiple clients M accelerates convergence through the increased number of participating clients. This property is particularly desirable in the federated setting, where large-scale parallelism can be leveraged to speed up training. In contrast, the third and final terms do not benefit from parallelism across clients. Nevertheless, their impact is mitigated by the faster convergence rate with respect to T , which decays faster than the dominant terms. The fourth term, which captures client heterogeneity and gradient variance at the client side, does not contain the τ acceleration factor. This further confirms that multiple local updates contribute to the acceleration of initial error and variance introduced by the server, while the client side does not benefit from it. More importantly, while the server-side learning rate decrease with τ , the global learning rate amplifies by τ . The intuition behind this is as follows: as the server side uses stale information to update, a smaller learning rate ensures that each server update remains close to the original model, preventing large deviations. However, smaller learning rates reduce the cumulative gradient step at the server. To ensure a globally faster convergence rate, the global aggregation compensates for this by applying a slightly larger learning rate. Finally, the overall convergence rate is $\mathcal{O}(\sqrt{d}/(\tau TM))$, demonstrating that multiple local updates τ and multiple clients M jointly accelerate convergence in SFL.

Straggler resilient communication time. The total communication time in SFL is largely determined by the straggler, as all other parties must wait for the slowest client to complete its computation before

Table 1: Test accuracy on four datasets. We run each method for 100 epochs on Fashion-MNIST and 500 epochs on the others, and report the resulting test accuracy at the final epoch.

Dataset	GAS	Vanilla SplitFed($\tau = 1$)	Ours($\tau = 2$)	Ours($\tau = 3$)	Ours($\tau = 4$)
CIFAR-10	75.28	69.73	77.86	73.20	69.40
Fashion-MNIST	83.70	77.50	85.45	85.28	84.47
CINIC-10	57.80	51.96	59.50	55.75	52.43
CIFAR-100	25.33	16.58	32.16	24.64	22.38

proceeding to the next communication round. We first define three terms for further explanation: 1) $t_{\text{straggler}}$ denotes the time delay of the straggler, 2) T_0 represents the number of communication rounds required for convergence, and 3) t_{server} as the server-side computation time for one local update. In parallel SFL settings, the required total delay caused by straggler can be represented as $T_0 \cdot t_{\text{straggler}}$, which mainly depends on the straggler and results in slow and unstable convergence.

In contrast, with unbalanced updates, if we let the server perform $\tau = t_{\text{straggler}}/t_{\text{server}}$ local iterations during each round. According to Corollary 4.4, this reduces the total number of communication rounds from T_0 to $T_1 = T_0/\tau$. Consequently, the total communication time becomes:

$$T_1 \cdot t_{\text{straggler}} = T_0 \cdot t_{\text{straggler}}/\tau = T_0 \cdot t_{\text{server}}, \quad (12)$$

which is now *independent of the straggler time*. This result highlights a key advantage of MU-SplitFed: by appropriately choosing τ , the system can effectively decouple overall training time from the performance of the slowest client.

5 Experiments

Experimental Setup. To evaluate the effectiveness of MU-SplitFed, we conduct experiments on four image classification benchmarks: Fashion-MNIST [31], CINIC-10 [32], CIFAR-10, and CIFAR-100 [33]. All experiments are carried out on a node with 3 NVIDIA A100 40GB GPUs. The model cut layer is denoted as L_c , where $L_c = n$ means the model is split after the n -th block. For these tasks, we adopt the AlexNet architecture, assessing the framework’s ability to mitigate the impact of stragglers. As AlexNet contains only 8 layers, it offers limited flexibility in exploring different splitting configurations. To further analyze the role of the unbalanced update ratio τ in controlling communication round, we extend our study to a large language model (LLM), OPT-1.3B [34], which has 24 transformer blocks and enables a broader range of splitting strategies. We evaluate its performance on the SST-2 dataset [35], a binary sentiment classification task, to examine the applicability of MU-SplitFed in the LLM domain.

We compare MU-SplitFed with vanilla SplitFed and GAS [8], a recent SFL method that addresses stragglers via asynchronous updates. Vanilla SplitFed serves as a baseline without straggler mitigation strategy. To simulate the device heterogeneity, we follow the simulation design of [8, 12]. In particular, we sample the computation time from an exponential distribution to represent different computation capacities across different clients. In our experiment, we train 10 clients in total with 50% partial partitioning for each global aggregation. For a fairness comparison, we modify both vanilla SplitFed and GAS to use ZO optimization, aligning them with MU-SplitFed’s gradient-free design. Additionally, we evaluate the convergence performance w.r.t to time unit of our simulation, providing a direct measure of each method’s performance to straggler-induced delays.

Impact of τ Selection. First, we investigate how the choice of τ impacts the performance of our proposed MU-SplitFed. We compare the accuracy from the same global communication round across different methods: we pull the result of the 100th epoch for Fashion-MNIST, and choose the 500th epoch result for the rest three datasets. As shown in Table 1, we compare the training accuracy with different values of the server iterations $\tau \in \{2, 3, 4\}$. Our method achieves the highest accuracy when $\tau = 2$, demonstrating its effectiveness in reducing communication round. However, increasing τ over 2 leads to a noticeable drop in accuracy. This observation aligns with our theoretical insights. Specifically, Corollary 4.2 suggests that the choice of τ is related to the parameter size of the client-side submodel $d_c = \sqrt{\frac{d}{\tau}}$, which is governed by the cut layer L_c . Given the structure of AlexNet, $L_c = 2$ is the only split type satisfied this setting without violating the constraint $L_c \geq 1$. Thereby, $\tau = 2$ corresponds to the optimal choice of server steps given fixed cutting strategy. Consequently, as τ exceeds this value, the mismatch between τ and splitting strategy contributed to the observed accuracy drop. Based on this insight, we use $\tau = 2$ for our method in the next experiment.

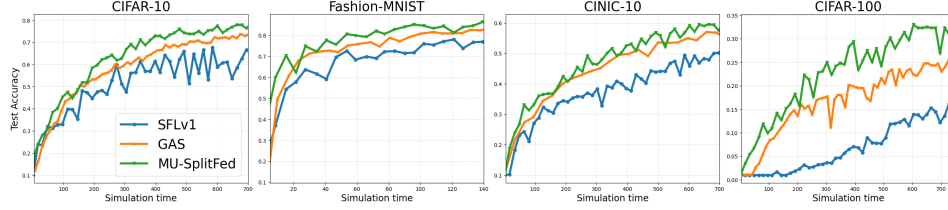


Figure 2: Performance Under Stragglers, where we set $\tau = 2$ for MU-SplitFed.

Performance under Straggler. In this subsection, we evaluate the resilience of MU-SplitFed to straggler effects by comparing its convergence performance against baseline methods on four datasets. Here, we introduce random delays following an exponential distribution to emulate straggler-induced latency. Figure 2 presents the accuracy over wall-clock time for all methods. Across all tasks, MU-SplitFed consistently achieves higher accuracy and in less time compared to both vanilla SplitFed and GAS, highlighting its efficiency in mitigating straggler-induced delays. Notably, on both CIFAR-10 and more complex task CIFAR-100, MU-SplitFed *maintains a fast and stable convergence trend, while GAS exhibits slower convergence and less consistency*. One possible reason for these scenarios is that GAS supports asynchronous updates, its activation generation step scales poorly with the increasing size of the label, introducing significant computational overhead that limits its efficiency in straggler-prone settings. In contrast, MU-SplitFed *maintains lightweight computation on both server and client sides, which allows efficient parallelization and better utilization of system resources during straggler delays*.

Interaction Between Cut Layer and Server Iterations. To fully explore the potential of our proposed unbalanced update in reducing the communication round, we fine-tune the OPT-1.3B that enables more types of model splitting. This allows us to more thoroughly explore how to jointly select τ and cut layer L_c to optimize communication efficiency. Figure 3 shows the total communication round required to attain 85% accuracy across different cut layers and values of τ . For a fixed cut layer (e.g. $L_c = 4$), increasing τ reduces communication round by up to 33%, confirming the benefit of unbalanced updates.

More interestingly, there a clear trade-off emerging between τ and L_c . When L_c is fixed, increasing τ initially improves convergence, but excessive server updates eventually lead to diminishing or adverse effects. Conversely, when fixing τ and tuning the cut layer, convergence consistently improves as L_c decreases, *indicating a deeper server-side model is beneficial for model performance*. Moreover, the optimal value of τ shifts higher as L_c moves earlier in the model. These trends confirm our theoretical insight in Remark 4.1: *to fully exploit server-side acceleration, the model partition must scale with the number of server iterations*. The dashed gray curve illustrates this joint optimization trajectory, highlighting that coordinated tuning of L_c and τ yields the most communication-efficient convergence.

Memory Efficiency. To evaluate the memory efficiency of our ZO-based framework in the context of LLM fine-tuning, we compare the peak memory usage on the client side. Specifically, we compare our proposed MU-SplitFed with FedAvg [4] and FedAvg with LoRA [36] (FedLoRA) for fine-tuning the OPT-1.3B model on the SST-2 dataset. As illustrated in Figure. 4, FedAvg incurs a peak memory cost of 8.02 GB on the client. FedLoRA, which reduces memory usage by updating only low-rank adapter matrices, reduces this to 5.64 GB. Despite these improvements, both FedAvg and FedLoRA still require substantial memory to store gradients and maintain the full model locally. In contrast,

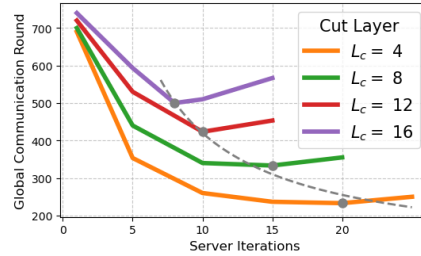


Figure 3: Interaction between cut layer L_c and server iteration τ .

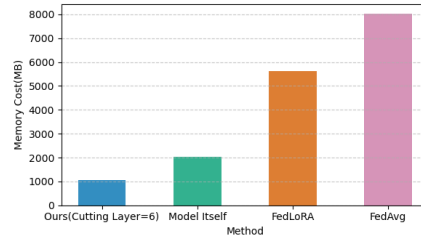


Figure 4: Comparison of peak memory cost for different methods for fine-tuning LLM.

MU-SplitFed reduces the peak client-side memory footprint to just 1.05 GB. This is achieved by storing only a partial model on the client and leveraging ZO optimization, which eliminates the need to store gradient information during training, further contributing to its memory efficiency.

6 Related Work

Split Federated Learning. SFL [1] is a powerful distributed learning framework that enables scalable training across resource-constrained edge devices. By model partitioning on the client side without sharing raw data with the server, SFL provides a memory-efficient and privacy-preserving solution for resource-constrained devices. Recent advances in SFL have addressed key challenges from different perspectives. To mitigate the communication bottleneck, Chen et al. [37] reduces communication frequency by proposing a loss threshold that determines when to exchange information between client and Split Server. Han et al. [3] employ different local loss functions on the client and server sides, thus reducing the gradient information transmission rounds. Other approaches apply quantization or sparsification techniques to reduce communication costs in each transaction round. For instance, [38] leverages Top-S sparsification for both forward embedding and backward gradient transmissions, while [39] introduces randomness for further enhancement. FedLite applies Top-K quantization to compress intermediate features [40]. For privacy purposes, several methods tackle model inversion attacks. ResSFL [41] and NoPeek [42] achieve attacker-aware training by integrating inversion score regularization term. Moreover, other strategies apply differential privacy on intermediate embedding features to provide privacy guarantees against label leakage [43]. In heterogeneous settings, methods like SCALA [44] and GAS [8] introduce activation concatenation and centralized training to enhance robustness and accommodate for varying client capabilities. However, theoretical research for SFL is still insufficient. [45] provides the first convergence analysis for sequential SFL, while [2] proposes an efficient update mechanism using different synchronization frequencies on client and server with rigorous convergence analysis for both sequential and parallel SFL.

Existing Straggler Solutions. The straggler issue in FL has been well explored, with asynchronous updates emerging as one of the most promising directions [10]. Yet, asynchronous methods rely on stale information to update, which can lead to performance degradation due to outdated or inconsistent model information. To address this, ASO-Fed [23] proposed a dynamic learning rate adjustment mechanism tailored to each client’s training progress to reduce the staleness effect from straggler. FedBuff [26] enables efficient training by using a buffer to store information from faster clients. Based on that, CA2FL [27] enhances the performance on heterogeneous data by adaptively adjusting model updates based on data property. Similarly, FedCompass [46] adopts a resource-aware scheduling policy that prioritizes clients with high computation capacity, thus mitigating the impact of stragglers. FedASMU [47] employs dynamic model aggregation with adaptive model adjustment to mitigate the impact of stragglers. Yet, existing strategies regarding the straggler in SFL remain limited. [9, 16] reduce the time delay by employing adaptive splitting strategies to balance the arrival times of activations. GAS [8] propose an asynchronous SFL framework that utilizes an activation buffer to generate activations based on the degree of bias, thereby enhancing the robustness of the algorithm.

7 Conclusion and Limitations

We propose MU-SplitFed, a simple and effective framework for mitigating the straggler problem in Split Federated Learning by introducing unbalanced updates on server-side. The simple yet efficient unbalanced update strategy enables faster training by reducing communication complexity, thereby mitigating delays caused by stragglers. Notably, both our theory and experiments show that increasing the unbalanced update ratio τ yields a linear reduction in communication frequency. When $\tau = t_{straggler}/t_{server}$, the total training time becomes independent of the straggler delay. Moreover, our analysis uncovers a key connection between the choice of the splitting layer and the optimal τ , offering practical guidance for further system design. These findings suggest that MU-SplitFed is a promising solution for enabling scalable and efficient training on resource-constrained edge devices.

Our work also highlights the potential of applying SFL for fine-tuning task of LLM, where memory efficiency is an impetus need. In LLM setting, SFL offers a natural fit: edge or local servers can serve as client-side device, while high-performance cloud servers act as the central server. Although our framework demonstrates initial promise in this direction by solving the bottleneck in this realm, how to fully realize the benefits of SFL for scalable LLM fine-tuning remains an open challenge and needs further investigation.

Acknowledgement

Research reported in this publication was supported by the National Institute Of General Medical Sciences of the National Institutes of Health under Award Numbers R16GM159671 and 1R35GM156653, and the National Science Foundation under Award Number 2045804. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

- [1] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, “Splitfed: When federated learning meets split learning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, 2022, pp. 8485–8493.
- [2] P. Han, C. Huang, G. Tian, M. Tang, and X. Liu, “Convergence analysis of split federated learning on heterogeneous data,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 103 476–103 544, 2024.
- [3] D.-J. Han, H. I. Bhatti, J. Lee, and J. Moon, “Accelerating federated learning with split learning on locally generated losses,” in *ICML 2021 workshop on federated learning for user privacy and data confidentiality. ICML Board*, 2021.
- [4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *AISTATS*, 2017.
- [5] P. Vepakomma, O. R. Gupta, A. Dubey, and R. Raskar, “Split learning for health: Distributed deep learning without sharing raw patient data,” *arXiv preprint arXiv:1812.00564*, 2018.
- [6] M. G. Poirrot, P. Vepakomma, K. Chang, J. Kalpathy-Cramer, R. Gupta, and R. Raskar, “Split learning for collaborative deep learning in healthcare,” *arXiv preprint arXiv:1912.12115*, 2019.
- [7] F. Mo, M. Malekzadeh, S. Chatterjee, F. Kawsar, and A. Mathur, “Enhancing efficiency in multidevice federated learning through data selection,” *arXiv preprint arXiv:2211.04175*, 2022.
- [8] J. Yang and Y. Liu, “Gas: Generative activation-aided asynchronous split federated learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, 2025, pp. 21 956–21 964.
- [9] D. Yan, M. Hu, Z. Xia, Y. Yang, J. Xia, X. Xie, and M. Chen, “Have your cake and eat it too: Toward efficient and accurate split federated learning,” *arXiv preprint arXiv:2311.13163*, 2023.
- [10] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, B. A. y Arcas, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data, S. Diggavi, H. Eichner, A. Gadhikar, Z. Garrett, A. M. Girgis, F. Hanzely, A. Hard, C. He, S. Horvath, Z. Huo, A. Ingerman, M. Jaggi, T. Javidi, P. Kairouz, S. Kale, S. P. Karimireddy, J. Konecny, S. Koyejo, T. Li, L. Liu, M. Mohri, H. Qi, S. J. Reddi, P. Richtarik, K. Singhal, V. Smith, M. Soltanolkotabi, W. Song, A. T. Suresh, S. U. Stich, A. Talwalkar, H. Wang, B. Woodworth, S. Wu, F. X. Yu, H. Yuan, M. Zaheer, M. Zhang, T. Zhang, C. Zheng, C. Zhu, and W. Zhu, “A field guide to federated optimization,” 2021. [Online]. Available: <https://arxiv.org/abs/2107.06917>
- [11] A. Hard, A. M. Girgis, E. Amid, S. Augenstein, L. McConnaughey, R. Mathews, and R. Anil, “Learning from straggler clients in federated learning,” *arXiv preprint arXiv:2403.09086*, 2024.
- [12] A. Reiszadeh, I. Tziotis, H. Hassani, A. Mokhtari, and R. Pedarsani, “Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity,” *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 2, pp. 197–205, 2022.
- [13] I. Wang, P. Nair, and D. Mahajan, “Fluid: Mitigating stragglers in federated learning using invariant dropout,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 73 258–73 273, 2023.
- [14] J. Park, D.-J. Han, M. Choi, and J. Moon, “Sageflow: Robust federated learning against both stragglers and adversaries,” *Advances in neural information processing systems*, vol. 34, pp. 840–851, 2021.
- [15] J. Tu, L. Yang, and J. Cao, “Distributed machine learning in edge computing: Challenges, solutions and future directions,” *ACM Computing Surveys*, vol. 57, no. 5, pp. 1–37, 2025.

- [16] J. Shen, N. Cheng, X. Wang, F. Lyu, W. Xu, Z. Liu, K. Aldubaikhy, and X. Shen, “Ringsfl: An adaptive split federated learning towards taming client heterogeneity,” *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 5462–5478, 2023.
- [17] S. Malladi, T. Gao, E. Nichani, A. Damian, J. D. Lee, D. Chen, and S. Arora, “Fine-tuning language models with just forward passes,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 53 038–53 075, 2023.
- [18] Z. Li, B. Ying, Z. Liu, C. Dong, and H. Yang, “Achieving dimension-free communication in federated learning via zeroth-order optimization,” in *The Thirteenth International Conference on Learning Representations*.
- [19] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney, “A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications,” *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 43–54, 2020.
- [20] H. Cai, Y. Lou, D. McKenzie, and W. Yin, “A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 1193–1203.
- [21] K. Nikolakakis, F. Haddadpour, D. Kalogerias, and A. Karbasi, “Black-box generalization: Stability of zeroth-order learning,” *Advances in neural information processing systems*, vol. 35, pp. 31 525–31 541, 2022.
- [22] S. Ghadimi and G. Lan, “Stochastic first-and zeroth-order methods for nonconvex stochastic programming,” *SIAM journal on optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.
- [23] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, “Asynchronous online federated learning for edge devices with non-iid data,” in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 15–24.
- [24] C. Xie, S. Koyejo, and I. Gupta, “Asynchronous federated optimization,” *arXiv preprint arXiv:1903.03934*, 2019.
- [25] Y. Chen, X. Sun, and Y. Jin, “Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, p. 4229–4238, Oct. 2020. [Online]. Available: <http://dx.doi.org/10.1109/TNNLS.2019.2953131>
- [26] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, “Federated learning with buffered asynchronous aggregation,” in *International conference on artificial intelligence and statistics*. PMLR, 2022, pp. 3581–3607.
- [27] Y. Wang, Y. Cao, J. Wu, R. Chen, and J. Chen, “Tackling the data heterogeneity in asynchronous federated learning with cached update calibration,” in *Federated learning and analytics in practice: algorithms, systems, applications, and opportunities*, 2023.
- [28] H. Yang, M. Fang, and J. Liu, “Achieving linear speedup with partial worker participation in non-iid federated learning,” *International Conference on Learning Representations*, 2021.
- [29] E. Chen, J. Zhang, S. Wang, C. Liu, and C. G. Brinton, “Parameter tracking in federated learning with adaptive optimization,” *CoRR*, 2025.
- [30] J. Zhang, E. Chen, C. Liu, and C. G. Brinton, “Dpzv: Resource efficient zo optimization for differentially private vfl,” *arXiv preprint arXiv:2502.20565*, 2025.
- [31] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.
- [32] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, “Cinic-10 is not imagenet or cifar-10,” *arXiv preprint arXiv:1810.03505*, 2018.
- [33] A. Krizhevsky, G. Hinton *et al.*, “Learning multiple layers of features from tiny images,” 2009.
- [34] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, “Opt: Open pre-trained transformer language models,” *arXiv preprint arXiv:2205.01068*, 2022.

- [35] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.
- [36] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, “Lora: Low-rank adaptation of large language models.” *ICLR*, vol. 1, no. 2, p. 3, 2022.
- [37] X. Chen, J. Li, and C. Chakrabarti, “Communication and computation reduction for split learning using asynchronous training,” in *2021 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, 2021, pp. 76–81.
- [38] B. Yuan, S. Ge, and W. Xing, “A federated learning framework for healthcare iot devices,” *arXiv preprint arXiv:2005.05083*, 2020.
- [39] F. Zheng, C. Chen, L. Lyu, and B. Yao, “Reducing communication for split learning by randomized top-k sparsification,” *arXiv preprint arXiv:2305.18469*, 2023.
- [40] J. Wang, H. Qi, A. S. Rawat, S. Reddi, S. Waghmare, F. X. Yu, and G. Joshi, “Fedlite: A scalable approach for federated learning on resource-constrained clients,” 2022. [Online]. Available: <https://arxiv.org/abs/2201.11865>
- [41] J. Li, A. S. Rakin, X. Chen, Z. He, D. Fan, and C. Chakrabarti, “Ressfl: A resistance transfer framework for defending model inversion attack in split federated learning,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 184–10 192.
- [42] P. Vepakomma, A. Singh, O. Gupta, and R. Raskar, “Nopeek: Information leakage reduction to share activations in distributed deep learning,” in *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2020, pp. 933–942.
- [43] D. Xiao, C. Yang, and W. Wu, “Mixing activations and labels in distributed training for split learning,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 3165–3177, 2022.
- [44] J. Yang and Y. Liu, “Scala: Split federated learning with concatenated activations and logit adjustments,” 2024. [Online]. Available: <https://arxiv.org/abs/2405.04875>
- [45] Y. Li and X. Lyu, “Convergence analysis of sequential split learning on heterogeneous data,” *arXiv preprint arXiv:2302.01633*, 2023.
- [46] Z. Li, P. Chaturvedi, S. He, H. Chen, G. Singh, V. Kindratenko, E. A. Huerta, K. Kim, and R. Madduri, “Fedcompass: Efficient cross-silo federated learning on heterogeneous client devices using a computing power aware scheduler,” 2024. [Online]. Available: <https://arxiv.org/abs/2309.14675>
- [47] J. Liu, J. Jia, T. Che, C. Huo, J. Ren, Y. Zhou, H. Dai, and D. Dou, “Fedasmu: Efficient asynchronous federated learning with dynamic staleness-aware model update,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 13 900–13 908.

NeurIPS Paper Checklist

1. Claims

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

Answer: [Yes]

Justification: The goal of our work is to solve the straggler issue under Split Federated setting. Due to the frequent communication between client and Split Server, leading to the training time overhead. To solve this problem, we propose an unbalanced update mechanism that presents advantage in controlling the communication frequency, consequently reduce the total time delay induced by straggler.

Guidelines:

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

2. Limitations

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

Answer: [Yes]

Justification: The limitation of this paper is placed at the end of the conclusion part.

Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

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

Answer: [Yes]

Justification: We claim that all the assumptions we make are minimal and common in distributed optimization literature. The proof details are attached in the appendix.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [Yes]

Justification: We conduct comprehensive experiments to testify our claims.

Guidelines:

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

- (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
- (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

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

Answer: [Yes]

Justification: We use open access dataset for evaluation the performance of our proposed framework. We will later put our code on github.

Guidelines:

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

6. Experimental setting/details

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

Answer: [Yes]

Justification: The experimental setup and results are included in the paper, and the choice of hyperparameters is put on the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

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

Answer: [No]

Justification: The advantage of our proposed method is very large, that we surpass the baselines under straggler. To verify our theory we only need to observe the trend align with our theory finds.

Guidelines:

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

8. Experiments compute resources

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

Answer: [Yes]

Justification: We mention this at the beginning of the experiment part.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: The research presented in the paper fully conforms to the NeurIPS Code of Ethics. All experiments were conducted with integrity and transparency, ensuring fairness, accountability, and respect for privacy.

Guidelines:

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

10. Broader impacts

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

Answer: [NA]

Justification: There is no societal impact of the work performed.

Guidelines:

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

11. Safeguards

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

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: We clearly cite every assets that we use in experiment section.

Guidelines:

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

13. New assets

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

Answer: [NA]

Justification: Our paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

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

Answer: [NA]

Justification: Our paper does not involve crowd-sourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

Answer: [NA]

Justification: Our paper does not involve crowd-sourcing nor research with human subjects.

Guidelines:

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

16. Declaration of LLM usage

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

Answer: [NA]

Justification: The core method of our work does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.

- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

A	Communication Benefits of Unbalanced Update	23
A.1	Dimension-Free ZOO achieved by Unbalanced Updates	23
A.2	Comparable Analysis	23
B	Preliminaries	24
B.1	Notations	24
B.2	Assumptions	24
B.3	Technical Lemmas	24
C	Proof For MU-Split	25
C.1	Proof of main theorem	25
C.1.1	One-Round Update on Server Side	25
C.1.2	One-Round Update on Client Side	27
C.1.3	Server-Client Combination	28
C.1.4	Justification for Corollary 4.2	29
C.2	Important Lemmas	29
D	Proof for for MU-SplitFed	32
D.1	Proof of main theorem	32
D.1.1	One-Round Update on Server Side	32
D.1.2	One-Round Update on Client Side	34
D.1.3	Server-Client Combination	35
D.1.4	Justification for Corollary 4.4	36
D.2	Important Lemmas	37
E	Additional Experiments	39
F	Choice of Hyperparameters	40

A Communication Benefits of Unbalanced Update

A.1 Dimension-Free ZOO achieved by Unbalanced Updates

As shown in Table 2, the proposed MU-SplitFed can further achieve dimension-free ZOO with convergence rate $\mathcal{O}(1/\sqrt{T})$, when $\tau \rightarrow d$. By appropriately scaling the unbalanced update factor τ to match the model dimension d , the convergence rate becomes independent of d . This is particularly significant for ZOO, where the parameter dimension d often dominates the denominator of the convergence rate and thus slows down training as the model size grows. Large models exacerbate this issue because the increased d in the denominator hinders convergence and adds communication overhead. MU-SplitFed mitigates this by exploiting unbalanced updates, which not only accelerates ZOO training but also reduces communication costs. Specifically, the convergence rate improves from $\mathcal{O}(\sqrt{d/T})$ to $\mathcal{O}(1/\sqrt{T})$, meanwhile, the communication complexity reduces from $\mathcal{O}(d/\epsilon^2)$ to $\mathcal{O}(1/\epsilon^2)$. Compared with other dimension-free methods [17, 18], which often rely on strong assumptions, e.g. low-rank assumption, that are impractical in real-world scenarios, MU-SplitFed provides a more flexible way towards this end. By introducing unbalanced updates into ZOO, we effectively remove the dependency on d without imposing additional assumptions, making the method significantly more feasible in practice.

A.2 Comparable Analysis

We analyze the communication costs of MU-SplitFed under different choices of τ and compare against two existing theoretical baselines for SFL frameworks. SFL-V1, introduced in [2], serves as the fundamental baseline for parallel SFL architectures using first-order optimization. Reference [45] provides rigorous convergence analysis with the perspective of SFL in a sequential update manner. To systematically validate the benefits of unbalanced updates, we present results across different τ configurations: $\tau = 1$ represents the balanced update scenario where client and server updates with equal frequency, providing insight into combining ZOO with traditional SFL; $\tau > 1$ corresponds to our proposed unbalanced update strategy; and $\tau \rightarrow d$ is the optimal case that τ scales to the same order of dimensionality d . That being said, the convergence rate is no longer dependent on d , thus achieving the dimension-free convergence rate.

Communication Advantage of Unbalanced Updates. Compared to balanced SFL with ZOO ($\tau = 1$), our unbalanced update strategy ($\tau > 1$) demonstrates linear convergence acceleration with respect to τ . This improvement translates directly to communication complexity, where τ provides linear communication cost reduction from $\mathcal{O}(d/M\epsilon^2)$ to $\mathcal{O}(d/\tau M\epsilon^2)$. Specifically, unbalanced updates reduce total communication overhead by decreasing the number of communication rounds required for convergence. When $\tau \rightarrow d$, we achieve a convergence rate of $\mathcal{O}(\sqrt{1/TM})$ that eliminates dependence on dimensionality d , resulting in dimension-free communication complexity of $\mathcal{O}(1/M\epsilon^2)$.

Comparison with SFL-V1. To the best of our knowledge, SFL-V1 [2] provides the first theoretical analysis for parallel SFL under bounded gradient, non-convex, and non-iid assumptions. However, their theoretical results exhibit no acceleration with respect to either the number of clients M or local update steps. In contrast, our convergence rate demonstrates faster convergence as both the number of clients M increase under the more loose assumption, e.g. bounded variance, consequently requiring fewer communication rounds to reach an ϵ -approximation solution.

Comparison with SFL-V2. Our method achieves comparable convergence rates to SFL-V2 [45], where K is the number of local updates. While multiple local updates K accelerate convergence

Method	Convergence Rate	SplitServer Comm. Cost	Assumptions
SFL-V1 [2]	$\mathcal{O}(1/\sqrt{T})$	$\mathcal{O}(K/\epsilon^2)$	b.g./N.C./non-iid
SFL-V2 [45]	$\mathcal{O}(1/\sqrt{TMK})$	$\mathcal{O}(K/M\epsilon^2)$	b.v./N.C./non-iid
MU-SplitFed ($\tau = 1$)	$\mathcal{O}(\sqrt{d/TM})$	$\mathcal{O}(d/M\epsilon^2)$	b.v./N.C./non-iid
MU-SplitFed ($\tau > 1$)	$\mathcal{O}(\sqrt{d/\tau TM})$	$\mathcal{O}(d/\tau M\epsilon^2)$	
MU-SplitFed ($\tau \rightarrow d$)	$\mathcal{O}(\sqrt{1/TM})$	$\mathcal{O}(1/M\epsilon^2)$	

Table 2: Comparison of Communication Complexity

in FL settings by reducing communication frequency, they impose additional communication costs when applied to SFL architectures. As demonstrated in Table 2, increasing local updates K actually increases the total communication cost for convergence in the SFL setting. This counterintuitive result stems from the relay-based update mechanism inherent in SFL, where local updates exacerbate communication overhead between clients and servers rather than reducing it. Conversely, our unbalanced update parameter τ facilitates convergence without requiring additional communication rounds, achieving linear communication cost reduction with respect to τ . This fundamental architectural advantage establishes the superior communication efficiency of our unbalanced update strategy over existing theoretical result.

B Preliminaries

B.1 Notations

Table 3: Notations in this paper

Notation	Meaning
d	Total model parameter dimension
m, M	Index, total number of clients
t, T	Index, total number of communication round
p, P	Index, total number of perturbations
i, τ	Index, total number of server iterations
\mathbf{x}^t	Global model parameters in the t -th round
x_c^t	Client-side model parameters in the t -th round
$x_s^{t,i}$	Server-side model parameters in the i -th iteration
ξ_m^t	Data sample in the t -th round for m -th client
$g_{c,p}^{t,i}$	Stochastic Zeroth-order gradient for t -th round
$g_{s,p}^{t,i}$	Stochastic Zeroth-order gradient for i -th iteration
$G_{c,m}^t$	Zeroth-order gradient estimator for client
$G_{s,m}^{t,i}$	Zeroth-order gradient estimator for server
$f_m(\cdot)$	Local loss function for client m
$f(\cdot)$	Global loss function for SL or SFL

B.2 Assumptions

Assumption B.1 (*L-Smooth*). For $\forall m \in [M]$, the loss function f_m is bounded from below, and is L -smooth, i.e. $\forall x, y, \|\nabla f_m(x) - \nabla f_m(y)\| \leq L\|x - y\|$.

Assumption B.2 (*Bounded variance*). For $\forall m \in [M]$, the variance of the stochastic gradient w.r.t. the client and the server is upper-bounded by σ_c^2 and σ_s^2 . Specifically, for $\forall \xi \in \mathcal{D}_m$,

$$\|\nabla_{x_c} F_m(x; \xi) - \nabla f_m(x)\|^2 \leq \sigma_c^2$$

$$\|\nabla_{x_s} F_m(x; \xi) - \nabla f_m(x)\|^2 \leq \sigma_s^2$$

Assumption B.3 (*Bounded Heterogeneity*). For $\forall m \in [M]$, the global variability of the local gradient is upper bounded:

$$\|\nabla f_m(x) - \nabla f(x)\|^2 \leq \epsilon^2$$

B.3 Technical Lemmas

Lemma B.1. Let $g(x)$ be defined as in (3). We define the smoothed function $f_\lambda(x) = \mathbb{E}_v[f(x + \lambda v)]$, where v is uniformly sampled from the Euclidean ball $\sqrt{d}\mathbb{B}^d = \{x \in \mathbb{R}^d \mid \|x\| \leq \sqrt{d}\}$. The following properties hold:

(i) $f_\lambda(x)$ is differentiable and $\mathbb{E}_u[g_\lambda(x)] = \nabla f_\lambda(x)$.

(ii) If $f(x)$ is L -smooth, then we have that

$$\|\nabla f(x) - \nabla f_\lambda(x)\| \leq \frac{L}{2} \lambda d^{3/2}, \quad (13)$$

and

$$\mathbb{E}_u[\|g_\lambda(x)\|^2] \leq 2d \cdot \|\nabla f(x)\|^2 + \frac{L^2}{2}\lambda^2 d^3. \quad (14)$$

Remark B.1. By (13) we immediately have

$$\|\nabla f_\lambda(x)\|^2 \leq 2\|\nabla f(x)\|^2 + \frac{L^2}{2}\lambda^2 d^3 \quad (15)$$

$$\|\nabla f(x)\|^2 \leq 2\|\nabla f_\lambda(x)\|^2 + \frac{L^2}{2}\lambda^2 d^3 \quad (16)$$

The dual-paced model aggregation and model update in SFL presents more challenge in convergence analysis compared to the analysis in traditional FL setting. To address this problem, we decompose the convergence analysis into client-side and server-side, respectively. The following lemma reveals this relationship.

Lemma B.2 (Decomposition). *Let $\mathbf{x}^t \equiv [x_c^t; x_s^t]$ denote the global model at the t th training rounds. By applying Assumption B.1, we have:*

$$\begin{aligned} & \mathbb{E}[f(\mathbf{x}^{t+1}) - f(\mathbf{x}^t)] \\ & \leq \mathbb{E}[\langle \nabla_{\mathbf{x}} f(\mathbf{x}^t), \mathbf{x}^{t+1} - \mathbf{x}^t \rangle] + \frac{L}{2} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \\ & \leq \underbrace{\mathbb{E}[\langle \nabla_{x_s} f(\mathbf{x}^t), x_s^{t+1} - x_s^t \rangle]}_{\mathcal{K}_1} + \underbrace{\frac{L}{2} \mathbb{E}[\|x_s^{t+1} - x_s^t\|^2]}_{\mathcal{K}_2} + \underbrace{\mathbb{E}[\langle \nabla_{x_c} f(\mathbf{x}^t), x_c^{t+1} - x_c^t \rangle]}_{\mathcal{K}_3} + \underbrace{\frac{L}{2} \mathbb{E}[\|x_c^{t+1} - x_c^t\|^2]}_{\mathcal{K}_4} \end{aligned} \quad (17)$$

C Proof For MU-Split

C.1 Proof of main theorem

We now prove the main theorem of MU-Split, and defer all important lemmas to Appendix C.2. We first restate the main theorem below.

Theorem C.1. *Under Assumption B.1 and B.2, and let the server iteration number be τ . If the learning rates satisfy $\eta_c/\tau = \eta_s = \eta \leq \min\{\frac{1}{64L(\tau+2d_s)}, \frac{1}{16L\tau d_c}\}$, the sequence of iterates generated by MU-Split satisfies:*

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^T \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] & \leq \frac{4}{\eta\tau T} \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)] + 16\eta L(\eta\tau L + 1)d_s\sigma_s^2 \\ & \quad + 8\eta\tau L d_c\sigma_c^2 + 4L^2(\eta^2\tau^2L^2 + 1/4)\lambda^2 d_s^3 + L^2\lambda^2 d_c^3, \end{aligned} \quad (18)$$

C.1.1 One-Round Update on Server Side

For \mathcal{K}_1 :

$$\begin{aligned} & \mathbb{E}[\langle \nabla_{x_s} f(\mathbf{x}^t), x_s^{t+1} - x_s^t \rangle] \\ & = \mathbb{E}[\langle \nabla_{x_s} f(\mathbf{x}^t), -\sum_{i=0}^{\tau-1} \eta_s G_s(\mathbf{x}^{t,i}; \xi^t) \rangle] \\ & = \mathbb{E}[\langle \nabla_{x_s} f(\mathbf{x}^t), -\sum_{i=0}^{\tau-1} \eta_s (\nabla_{x_s} f_\lambda^{t,i} - \nabla_{x_s} f(\mathbf{x}^t) + \nabla_{x_s} f(\mathbf{x}^t)) \rangle] \\ & = \mathbb{E}[\langle \sqrt{\eta_s\tau} \nabla_{x_s} f(\mathbf{x}^t), -\frac{\sqrt{\eta_s}}{\sqrt{\tau}} \sum_{i=0}^{\tau-1} (\nabla_{x_s} f_\lambda^{t,i} - \nabla_{x_s} f(\mathbf{x}^t)) \rangle] - \eta_s\tau \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] \\ & = \frac{\eta_s\tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_s}{2\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} (\nabla_{x_s} f_\lambda^{t,i} - \nabla_{x_s} f(\mathbf{x}^t)) \right\|^2 \\ & \quad - \frac{\eta_s}{2\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_\lambda^{t,i} \right\|^2 - \eta_s\tau \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] \end{aligned}$$

$$\begin{aligned}
&= -\frac{\eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_s}{2\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} \left(\nabla_{x_s} f_{\lambda}^{t,i} - \nabla_{x_s} f^{t,i} + \nabla_{x_s} f^{t,i} - \nabla_{x_s} f^t \right) \right\|^2 \\
&\quad - \frac{\eta_s}{2\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{\lambda}^{t,i} \right\|^2 \\
&\leq -\frac{\eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_s}{\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} (\nabla_{x_s} f_{\lambda}^{t,i} - \nabla_{x_s} f^{t,i}) \right\|^2 + \frac{\eta_s}{\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} (\nabla_{x_s} f_m^{t,i} - \nabla_{x_s} f_m^t) \right\|^2 \\
&\quad - \frac{\eta_s}{2\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{\lambda}^{t,i} \right\|^2 \\
&\leq -\frac{\eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \eta_s \sum_{i=0}^{\tau-1} \mathbb{E} \left\| \nabla_{x_s} f_{\lambda}^{t,i} - \nabla_{x_s} f^{t,i} \right\|^2 \\
&\quad + \eta_s \sum_{i=0}^{\tau-1} \mathbb{E} \left\| \nabla_{x_s} f^{t,i} - \nabla_{x_s} f^t \right\|^2 - \frac{\eta_s}{2\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{\lambda}^{t,i} \right\|^2 \\
&\leq -\frac{\eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_s}{4} \tau L^2 \lambda^2 d_s^3 + \underbrace{\eta_s L^2 \sum_{i=0}^{\tau-1} \mathbb{E} \|x_s^{t,i} - x_s^t\|^2}_{\mathcal{A}_1} - \frac{\eta_s}{2\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{\lambda}^{t,i} \right\|^2 \\
&\leq -\frac{\eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_s}{4} \tau L^2 \lambda^2 d_s^3 - \frac{\eta_s}{2\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{\lambda}^{t,i} \right\|^2 \\
&\quad 2\eta_s L^2 (8\eta_s^2(\tau^3 + \tau^2 d_s/P) \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 4\eta_s^2 \tau^2 d_s \sigma_s^2/P + \eta_s^2 \tau^3 L^2 \lambda^2 d_s^3) \\
&= \left(16\eta_s^3 L^2 (\tau^3 + \tau^2 d_s/P) - \frac{\eta_s \tau}{2} \right) \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_s}{4} \tau L^2 \lambda^2 d_s^3 \\
&\quad + \frac{8\eta_s^3 \tau^2 L^2 d_s \sigma_s^2}{P} + 2\eta_s^3 \tau^3 L^4 \lambda^2 d_s^3 - \frac{\eta_s}{2\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{\lambda}^{t,i} \right\|^2 \tag{19}
\end{aligned}$$

where we apply L -smooth, Lemma B.1, $\langle a, b \rangle \leq \frac{\|a\|^2 + \|b\|^2}{2}$ and $\|a + b\|^2 \leq 2(\|a\|^2 + \|b\|^2)$, and substitute Lemma C.4 into \mathcal{A}_1 .

For \mathcal{K}_2 :

$$\mathbb{E}[\|x_s^{t+1} - x_s^t\|^2] = \eta_s^2 \mathbb{E} \left[\left\| \sum_{i=0}^{\tau-1} G_s(\mathbf{x}^{t,i}; \xi^t) \right\|^2 \right]$$

By (31):

$$\begin{aligned}
&\mathbb{E} \left[\left\| \sum_{i=0}^{\tau-1} G_s(\mathbf{x}^{t,i}; \xi^t) \right\|^2 \right] \\
&\leq 2\mathbb{E} \left[\left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{\lambda}(\mathbf{x}^{t,i}) \right\|^2 \right] + 2 \sum_{i=0}^{\tau-1} \mathbb{E} \left[\left\| G_s(\mathbf{x}^{t,i}; \xi^t) - \nabla_{x_s} f_{\lambda}(\mathbf{x}^{t,i}) \right\|^2 \right]
\end{aligned}$$

Similar to the proof in C.4, we substitute in (25) and (30) in order:

$$\begin{aligned}
&\sum_{i=0}^{\tau-1} \mathbb{E} \left[\left\| G_s(\mathbf{x}^{t,i}; \xi^t) - \nabla_{x_s} f_{\lambda}(\mathbf{x}^{t,i}) \right\|^2 \right] \\
&\leq \frac{-1}{P} \sum_{i=0}^{\tau-1} \mathbb{E} \left[\left\| \nabla_{x_s} f_{\lambda}^{t,i} \right\|^2 \right]
\end{aligned}$$

$$\begin{aligned}
& + \sum_{i=0}^{\tau-1} \left(\frac{4d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{4L^2 d_s}{P} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] + \frac{2d_s \sigma_s^2}{P} + \frac{L^2 \lambda^2 d_s^3}{2P} \right) \\
& \leq \frac{-1}{P} \sum_{i=0}^{\tau-1} \left(4\mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 4L^2 \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] + \frac{L^2}{2} \lambda^2 d_s^3 \right) \\
& \quad + \sum_{i=0}^{\tau-1} \left(\frac{4d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{4L^2 d_s}{P} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] + \frac{2d_s \sigma_s^2}{P} + \frac{L^2 \lambda^2 d_s^3}{2P} \right) \\
& \leq \frac{4\tau d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{4\tau L^2 d_s}{P} \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] + \frac{2\tau d_s \sigma_s^2}{P}
\end{aligned}$$

So

$$\begin{aligned}
\mathbb{E}[\|x_s^{t+1} - x_s^t\|^2] & \leq \frac{8\eta_s^2 \tau d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{8\eta_s^2 \tau L^2 d_s}{P} \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] \\
& \quad + \frac{4\eta_s^2 \tau d_s \sigma_s^2}{P} + 2\eta_s^2 \mathbb{E}[\|\sum_{i=0}^{\tau-1} \nabla_{x_s} f_\lambda(\mathbf{x}^{t,i})\|^2] \\
(1 - \frac{8\eta_s^2 \tau L^2 d_s}{P}) \mathbb{E}[\|x_s^{t+1} - x_s^t\|^2] & \leq \frac{8\eta_s^2 \tau d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{4\eta_s^2 \tau d_s \sigma_s^2}{P} \\
& \quad + 2\eta_s^2 \mathbb{E}[\|\sum_{i=0}^{\tau-1} \nabla_{x_s} f_\lambda(\mathbf{x}^{t,i})\|^2]
\end{aligned}$$

Further assume that $\eta_s \leq \frac{1}{4L\sqrt{\tau d_s/P}}$, we have

$$\mathbb{E}[\|x_s^{t+1} - x_s^t\|^2] \leq \frac{16\eta_s^2 \tau d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{8\eta_s^2 \tau d_s \sigma_s^2}{P} + 4\eta_s^2 \mathbb{E}[\|\sum_{i=0}^{\tau-1} \nabla_{x_s} f_\lambda(\mathbf{x}^{t,i})\|^2] \quad (20)$$

C.1.2 One-Round Update on Client Side

For \mathcal{K}_3 , we have

$$\begin{aligned}
& \mathbb{E}\langle \nabla_{x_c} f(\mathbf{x}^t), x_c^{t+1} - x_c^t \rangle \\
& = \mathbb{E}\langle \nabla_{x_c} f(\mathbf{x}^t), \eta_c G_c(\mathbf{x}^t; \xi^t) \rangle \\
& = \mathbb{E}\langle \nabla_{x_c} f(\mathbf{x}^t), -\eta_c (\nabla_{x_c} f_\lambda^t - \nabla_{x_c} f(\mathbf{x}^t) + \nabla_{x_c} f(\mathbf{x}^t)) \rangle \\
& = \mathbb{E}\langle \sqrt{\eta_c} \nabla_{x_c} f(\mathbf{x}^t), -\sqrt{\eta_c} (\nabla_{x_c} f_\lambda^t - \nabla_{x_c} f^t) \rangle - \eta_c \mathbb{E}\|\nabla_{x_c} f(\mathbf{x}^t)\|^2 \\
& = \frac{\eta_c}{2} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{\eta_c}{2} \mathbb{E}[\|(\nabla_{x_c} f_\lambda^t - \nabla_{x_c} f^t)\|^2] - \frac{\eta_c}{2} \mathbb{E}\|\nabla_{x_c} f_\lambda^t\|^2 - \eta_c \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] \\
& \leq -\frac{\eta_c}{2} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{\eta_c}{4} L^2 \lambda^2 d_c^3 - \frac{\eta_c}{2} \mathbb{E}\|\nabla_{x_c} f_\lambda^t\|^2 \quad (21)
\end{aligned}$$

For \mathcal{K}_4 :

$$\begin{aligned}
\mathbb{E}[\|x_c^{t+1} - x_c^t\|^2] & = \eta_c^2 \mathbb{E}\|G_c^t(\mathbf{x}^t; \xi^t)\|^2 \\
& = \eta_c^2 \mathbb{E}\|\nabla_{x_c} f_\lambda^t\|^2 + \eta_c^2 \mathbb{E}\|G_c^t - \nabla_{x_c} f_\lambda^t\|^2
\end{aligned}$$

Substituting (25) and (30) in order, we have

$$\begin{aligned}
& \eta_c^2 \mathbb{E}\|G_c^t - \nabla_{x_c} f_\lambda^t\|^2 \\
& \leq \eta_c^2 \left(\frac{4d_c}{P} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{4L^2 d_c}{P} \mathbb{E}[\|x_c^{t+1} - x_c^t\|^2] + \frac{2d_c \sigma_c^2}{P} + \frac{L^2 \lambda^2 d_c^3}{2P} - \frac{1}{P} \mathbb{E}\|\nabla_{x_s} f_\lambda^t\|^2 \right) \\
& \leq \eta_c^2 \left(\frac{4d_c}{P} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{4L^2 d_c}{P} \mathbb{E}[\|x_c^{t+1} - x_c^t\|^2] + \frac{2d_c \sigma_c^2}{P} + \frac{L^2 \lambda^2 d_c^3}{2P} \right) \\
& \quad - \frac{\eta_c^2}{P} \left(4\mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + 4L^2 \mathbb{E}[\|x_c^{t,i} - x_c^t\|^2] + \frac{L^2}{2} \lambda^2 d_c^3 \right)
\end{aligned}$$

$$\leq \frac{4\eta_c^2 d_c}{P} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{4\eta_c^2 L^2 d_c}{P} \mathbb{E}[\|x_c^{t+1} - x_c^t\|^2] + \frac{2\eta_c^2 d_c \sigma_c^2}{P}$$

So

$$\begin{aligned} \mathbb{E}[\|x_c^{t+1} - x_c^t\|^2] &\leq \eta_c^2 \mathbb{E} \|\nabla_{x_c} f_\lambda^t\|^2 + \frac{4\eta_c^2 d_c}{P} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] \\ &\quad + \frac{4\eta_c^2 L^2 d_c}{P} \mathbb{E}[\|x_c^{t+1} - x_c^t\|^2] + \frac{2\eta_c^2 d_c \sigma_c^2}{P} \\ (1 - \frac{4\eta_c^2 L^2 d_c}{P}) \mathbb{E}[\|x_c^{t+1} - x_c^t\|^2] &\leq \eta_c^2 \mathbb{E} \|\nabla_{x_c} f_\lambda^t\|^2 + \frac{4\eta_c^2 d_c}{P} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{2\eta_c^2 d_c \sigma_c^2}{P} \end{aligned}$$

Further assume $\eta_c \leq \frac{1}{L\sqrt{8d_c/P}}$, and we have

$$\mathbb{E}[\|x_c^{t+1} - x_c^t\|^2] \leq 2\eta_c^2 \mathbb{E} \|\nabla_{x_c} f_\lambda^t\|^2 + \frac{8\eta_c^2 d_c}{P} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{4\eta_c^2 d_c \sigma_c^2}{P} \quad (22)$$

C.1.3 Server-Client Combination

We now substitute (19), (20), (21), (22) into (17):

$$\begin{aligned} &\mathbb{E}[f(\mathbf{x}^{t+1}) - f(\mathbf{x}^t)] \\ &\leq \underbrace{\mathbb{E}[\langle \nabla_{x_s} f(\mathbf{x}^t), x_s^{t+1} - x_s^t \rangle]}_{\kappa_1} + \underbrace{\frac{L}{2} \mathbb{E}[\|x_s^{t+1} - x_s^t\|^2]}_{\kappa_2} + \underbrace{\mathbb{E}[\langle \nabla_{x_c} f(\mathbf{x}^t), x_c^{t+1} - x_c^t \rangle]}_{\kappa_3} + \underbrace{\frac{L}{2} \mathbb{E}[\|x_c^{t+1} - x_c^t\|^2]}_{\kappa_4} \\ &\stackrel{(i)}{\leq} \underbrace{\left(16\eta_s^3 L^2 (\tau^3 + \tau^2 d_s/P) - \frac{\eta_s \tau}{2}\right) \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_s}{4} \tau L^2 \lambda^2 d_s^3 + \frac{8\eta_s^3 \tau^2 L^2 d_s \sigma_s^2}{P} + 2\eta_s^3 \tau^3 L^4 \lambda^2 d_s^3}_{\kappa_1} \\ &\quad - \underbrace{\frac{\eta_s}{2\tau} \mathbb{E} \left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_\lambda^{t,i} \right\|^2}_{\kappa_1} + \underbrace{\frac{8\eta_s^2 L \tau d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{4\eta_s^2 L \tau d_s \sigma_s^2}{P} + 2\eta_s^2 L \mathbb{E} \left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_\lambda(\mathbf{x}^{t,i}) \right\|^2}_{\kappa_2} \\ &\quad - \underbrace{\frac{\eta_c}{2} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{\eta_c}{4} L^2 \lambda^2 d_c^3 - \frac{\eta_c}{2} \mathbb{E} \|\nabla_{x_c} f_\lambda^t\|^2}_{\kappa_3} \\ &\quad + \underbrace{\eta_c^2 L \mathbb{E} \|\nabla_{x_c} f_\lambda^t\|^2 + \frac{4\eta_c^2 L d_c}{P} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{2\eta_c^2 L d_c \sigma_c^2}{P}}_{\kappa_4} \\ &\stackrel{(ii)}{\leq} \left(16\eta_s^2 \tau L (\tau + d_s/P) - \frac{\eta_s \tau}{2}\right) \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{4\eta_s^2 \tau L (2\eta_s \tau L + 1) d_s \sigma_s^2}{P} \\ &\quad + \eta_s \tau L^2 (2\eta_s^2 \tau^2 L^2 + 1/4) \lambda^2 d_s^3 + \left(\frac{4\eta_s^2 L d_c}{P} - \frac{\eta_c}{2}\right) \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{\eta_c}{4} L^2 \lambda^2 d_c^3 + \frac{2\eta_c^2 L d_c \sigma_c^2}{P} \\ &\stackrel{(iii)}{\leq} -\frac{\eta_s \tau}{4} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{4\eta_s^2 \tau L (2\eta_s \tau L + 1) d_s \sigma_s^2}{P} + \eta_s \tau L^2 (2\eta_s^2 \tau^2 L^2 + 1/4) \lambda^2 d_s^3 \\ &\quad - \frac{\eta_c}{4} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{\eta_c}{4} L^2 \lambda^2 d_c^3 + \frac{2\eta_c^2 L d_c \sigma_c^2}{P} \quad (23) \end{aligned}$$

where in (i) we applied (19), (20), (21), (22); in (ii) we assume $\eta_s \leq \frac{1}{\tau L}$ to index on terms of η_s , assume $\eta_s \leq \frac{1}{4\tau L}, \eta_c \leq \frac{1}{2L}$ to remove the term $\mathbb{E} \left\| \sum_{i=0}^{\tau-1} \nabla_{x_s} f_\lambda^{t,i} \right\|^2$, and combine the terms of $\|\nabla_{x_c} f(\mathbf{x}^t)\|^2$ and $\|\nabla_{x_c} f(\mathbf{x}^t)\|^2$. In (iii), we let

$$\eta_s \leq \frac{P}{64L(\tau P + 2d_s)}$$

And

$$\eta_c \leq \frac{P}{16Ld_c}$$

To combine the squared norm of the server gradient $\mathbb{E}[\|\nabla_{x_s} f\|^2]$ and client gradient $\mathbb{E}[\|\nabla_{x_c} f\|^2]$, we define the universal step size $\eta := \eta_s$, and let $\eta_c = \eta\tau$. Rearranging the terms in (23), we have

$$\begin{aligned} \frac{\eta\tau}{4} (\mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2]) &\leq \mathbb{E}[f(\mathbf{x}^t) - f(\mathbf{x}^{t+1})] + \frac{4\eta^2\tau L(2\eta\tau L + 1)d_s\sigma_s^2}{P} \\ &\quad + \eta\tau L^2(2\eta^2\tau^2 L^2 + 1/4)\lambda^2 d_s^3 + \frac{\eta}{4}\tau L^2\lambda^2 d_c^3 \\ &\quad + \frac{2\eta^2\tau^2 L d_c\sigma_c^2}{P} \\ \frac{\eta\tau}{4} \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] &\leq \mathbb{E}[f(\mathbf{x}^t) - f(\mathbf{x}^{t+1})] + \frac{4\eta^2\tau L(2\eta\tau L + 1)d_s\sigma_s^2}{P} \\ &\quad + \eta\tau L^2(2\eta^2\tau^2 L^2 + 1/4)\lambda^2 d_s^3 + \frac{\eta}{4}\tau L^2\lambda^2 d_c^3 + \frac{2\eta^2\tau^2 L d_c\sigma_c^2}{P} \end{aligned} \quad (24)$$

Take the average from $t = 0$ to $T - 1$ at both sides:

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \frac{\eta\tau}{4} \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] &\leq \frac{1}{T} \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)] + \frac{4\eta^2\tau L(2\eta\tau L + 1)d_s\sigma_s^2}{P} \\ &\quad + \eta\tau L^2(2\eta^2\tau^2 L^2 + 1/4)\lambda^2 d_s^3 + \frac{\eta}{4}\tau L^2\lambda^2 d_c^3 + \frac{2\eta^2\tau^2 L d_c\sigma_c^2}{P} \\ \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] &\leq \frac{4}{\eta\tau T} \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)] + \frac{16\eta L(2\eta\tau L + 1)d_s\sigma_s^2}{P} \\ &\quad + 4L^2(2\eta^2\tau^2 L^2 + 1/4)\lambda^2 d_s^3 + L^2\lambda^2 d_c^3 + \frac{8\eta\tau L d_c\sigma_c^2}{P}, \end{aligned}$$

where in the last step we divided both sides by $\frac{\eta\tau}{4}$. Let $P = 1$, and we complete the proof.

C.1.4 Justification for Corollary 4.2

To further simplify the result and achieve the optimal convergence rate in Corollary 4.2, again, we assume $\eta \leq \frac{1}{\tau L}$. We also optimize upon η to get the convergence rate. Let $\eta = \frac{1}{\sqrt{d\tau T}}$, we derive that

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] \leq \frac{4\sqrt{d}}{\sqrt{\tau T}} \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)] + \frac{48Ld_s\sigma_s^2}{\sqrt{d\tau T}} + 9L^2\lambda^2 d_s^3 + L^2\lambda^2 d_c^3 + \frac{8\sqrt{\tau} L d_c\sigma_c^2}{\sqrt{d\tau T}}$$

Let $d_c = d/\sqrt{\tau}$ and $d_s = d - d/\sqrt{\tau}$, and further let

$$\lambda^2 = \frac{1}{\sqrt{\tau T} d^{5/2} L}$$

Thus, we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] \leq \frac{4\sqrt{d}}{\sqrt{\tau T}} \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)] + \frac{48L\sqrt{d}\sigma_s^2}{\sqrt{\tau T}} + \frac{9\sqrt{d}}{\sqrt{\tau T}} + \frac{8L\sigma_c^2}{\sqrt{\tau T}}$$

The convergence rate is seen to be $\mathcal{O}(\frac{\sqrt{d}}{\sqrt{\tau T}})$

C.2 Important Lemmas

Lemma C.2 (Bounds on the variance of Zeroth-order Gradient). *Under the same condition as Lemma B.1, and consider the stochastic Zeroth-order Gradient, we can further bound the variance of the stochastic Zeroth-order Gradient by true gradient at the beginning of the local iteration and the local update distance.*

$$\mathbb{E}[\|G_s^{t,i}(x_c^t, x_s^{t,i}; \xi^t) - \nabla_{x_s} f_\lambda(x_c^t, x_s^{t,i})\|^2] \leq \frac{4d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{4L^2 d_s}{P} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2]$$

$$+ \frac{2d_s\sigma_s^2}{P} + \frac{L^2\lambda^2d_s^3}{2P} - \frac{1}{P}\mathbb{E}[\|\nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})\|^2] \quad (25)$$

proof:

We use multi-perturbation to calculate the Zeroth-Order Oracle: $G_s^{t,i}(x_c^t, x_s^{t,i}; \xi^t) = \frac{1}{P} \sum_{p=1}^P g_{s,p}^{t,i}(x_c^t, x_s^{t,i}; \xi^t)$, where $g_{s,p}^{t,i}$ is the stochastic Zeroth-Order Oracle for one perturbation. Then, the λ -smooth function is represented as $\mathbb{E}_{u_p, \xi^t}[g_{s,p}^{t,i}(x_c^t, x_s^{t,i}; \xi^t)] = \nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})$. By Lemma B.1, we have

$$\mathbb{E}_u[\|g_{s,p}^{t,i}(x_c^t, x_s^{t,i}; \xi^t)\|^2] \leq 2d_s \cdot \|\nabla_{x_s}F(x_c^t, x_s^{t,i}; \xi^t)\|^2 + \frac{L^2}{2}\lambda^2d_s^3.$$

Thus we have

$$\begin{aligned} & \mathbb{E}[\|G_s^{t,i} - \nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})\|^2] \\ &= \frac{1}{P^2} \sum_{p=1}^P \mathbb{E}[\|g_{s,p}^{t,i}(x_c^t, x_s^{t,i}) - \nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})\|^2] \\ &= \frac{1}{P^2} \sum_{p=1}^P \mathbb{E}[\|g_{s,p}^{t,i}(x_c^t, x_s^{t,i})\|^2] - \frac{1}{P} \|\nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})\|^2 \\ &\leq \frac{1}{P^2} \sum_{p=1}^P \left[2d_s \mathbb{E}[\|\nabla_{x_s}F(x_c^t, x_s^{t,i}; \xi^t)\|^2] + \frac{L^2}{2}\lambda^2d_s^3 \right] - \frac{1}{P} \mathbb{E}[\|\nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})\|^2] \\ &\leq \frac{1}{P^2} \sum_{p=1}^P \left[2d_s (\mathbb{E}[\|\nabla_{x_s}f(x_c^t, x_s^{t,i})\|^2] + \sigma_s^2) + \frac{L^2}{2}\lambda^2d_s^3 \right] - \frac{1}{P} \mathbb{E}[\|\nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})\|^2] \\ &= \frac{1}{P} \left[2d_s \mathbb{E}[\|\nabla_{x_s}f(x_c^t, x_s^{t,i})\|^2] + 2d_s\sigma_s^2 + \frac{L^2}{2}\lambda^2d_s^3 - \mathbb{E}[\|\nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})\|^2] \right] \quad (26) \end{aligned}$$

The bound for the squared norm of the variance is:

$$\begin{aligned} \mathbb{E}[\|\nabla_{x_s}f(x_c^t, x_s^{t,i})\|^2] &= \mathbb{E}[\|\nabla_{x_s}f(x_c^t, x_s^{t,i}) - \nabla_{x_s}f(\mathbf{x}^t) + \nabla_{x_s}f(\mathbf{x}^t)\|^2] \\ &\leq 2\mathbb{E}[\|\nabla_{x_s}f(x_c^t, x_s^{t,i}) - \nabla_{x_s}f(\mathbf{x}^t)\|^2] + 2\mathbb{E}[\|\nabla_{x_s}f(\mathbf{x}^t)\|^2] \\ &\leq 2L^2\mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] + 2\mathbb{E}[\|\nabla_{x_s}f(\mathbf{x}^t)\|^2] \quad (27) \end{aligned}$$

Substituting (27) into (26), and we finish the proof.

Lemma C.3 (Bounds on the norm of the Zeroth-order gradient estimator).

$$\begin{aligned} & \mathbb{E}[\|G_s^{t,i}(x_c^t, x_s^{t,i})\|^2] \\ &\leq \frac{4(d_s + P - 1)}{P} \mathbb{E}[\|\nabla_{x_s}f(\mathbf{x}^t)\|^2] + \frac{4L^2(d_s + P - 1)}{P} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] + \frac{2d_s\sigma_s^2}{P} + \frac{L^2\lambda^2d_s^3}{2} \quad (28) \end{aligned}$$

proof:

It follows that

$$\mathbb{E}[\|G_s^{t,i}(x_c^t, x_s^{t,i})\|^2] = \mathbb{E}[\|G_s^{t,i}(x_c^t, x_s^{t,i}) - \nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})\|^2] + \mathbb{E}[\|\nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})\|^2] \quad (29)$$

From Lemma B.1 we have

$$\begin{aligned} \mathbb{E}[\|\nabla_{x_s}f_\lambda^t(x_c^t, x_s^{t,i})\|^2] &\leq 2\mathbb{E}[\|\nabla_{x_s}f(x_c^t, x_s^{t,i})\|^2] + \frac{L^2}{2}\lambda^2d_s^3 \\ &\leq 2\mathbb{E}[\|\nabla_{x_s}f(x_c^t, x_s^{t,i}) - \nabla_{x_s}f(\mathbf{x}^t) + \nabla_{x_s}f(\mathbf{x}^t)\|^2] + \frac{L^2}{2}\lambda^2d_s^3 \\ &\leq 4\mathbb{E}[\|\nabla_{x_s}f(x_c^t, x_s^{t,i}) - \nabla_{x_s}f(\mathbf{x}^t)\|^2] + 4\mathbb{E}[\|\nabla_{x_s}f(\mathbf{x}^t)\|^2] + \frac{L^2}{2}\lambda^2d_s^3 \\ &\leq 4\mathbb{E}[\|\nabla_{x_s}f(\mathbf{x}^t)\|^2] + 4L^2\mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] + \frac{L^2}{2}\lambda^2d_s^3 \quad (30) \end{aligned}$$

Then we can finish the proof by combining Lemma C.2 and (30).

Lemma C.4 (Bounds on multiple update steps(Zerth Order)). *If $\eta_s \leq \frac{\sqrt{P}}{4\tau L\sqrt{(P+d_s/\tau)}}$, we have*

$$\sum_{i=0}^{\tau-1} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] \leq \frac{16\eta_s^2\tau^3(P+d_s/\tau)}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{8\eta_s^2\tau^2\sigma_L^2 d_s}{P} + 2\eta_s^2\tau^3 L^2 \lambda^2 d_s^3$$

proof:

We first apply the update formula:

$$\sum_{i=0}^{\tau-1} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] = \sum_{i=0}^{\tau-1} \eta_s^2 \mathbb{E}[\|\sum_{j=0}^{i-1} G_s^{t,j}(x_c^t, x_s^{t,j})\|^2]$$

By the property of martingale difference sequence, we have

$$\begin{aligned} & \mathbb{E}[\|\sum_{j=0}^{i-1} G_s^{t,j}(x_c^t, x_s^{t,j})\|^2] \\ & \leq 2\mathbb{E}[\|\sum_{j=0}^{i-1} \nabla_{x_s} f_\lambda^t(x_c^t, x_s^{t,j})\|^2] + 2\mathbb{E}[\|\sum_{j=0}^{i-1} G_s^{t,j}(x_c^t, x_s^{t,j}) - \nabla_{x_s} f_\lambda^t(x_c^t, x_s^{t,j})\|^2] \\ & \leq 2i \sum_{j=0}^{i-1} \mathbb{E}[\|\nabla_{x_s} f_\lambda^t(x_c^t, x_s^{t,j})\|^2] + 2 \sum_{j=0}^{i-1} \mathbb{E}[\|G_s^{t,j}(x_c^t, x_s^{t,j}) - \nabla_{x_s} f_\lambda^t(x_c^t, x_s^{t,j})\|^2] \end{aligned} \quad (31)$$

We thus have

$$\begin{aligned} & \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] \\ & \leq 2 \sum_{i=0}^{\tau-1} \eta_s^2 \left(i \sum_{j=0}^{i-1} \mathbb{E}[\|\nabla_{x_s} f_\lambda^t(x_c^t, x_s^{t,j})\|^2] + \sum_{j=0}^{i-1} \mathbb{E}[\|G_s^{t,j}(x_c^t, x_s^{t,j}) - \nabla_{x_s} f_\lambda^t(x_c^t, x_s^{t,j})\|^2] \right) \\ & \leq 2\eta_s^2\tau^2 \sum_{i=0}^{\tau-1} \mathbb{E}[\|\nabla_{x_s} f_\lambda^t(x_c^t, x_s^{t,i})\|^2] + 2\eta_s^2\tau \sum_{i=0}^{\tau-1} \mathbb{E}[\|G_s^{t,i}(x_c^t, x_s^{t,i}) - \nabla_{x_s} f_\lambda^t(x_c^t, x_s^{t,i})\|^2], \end{aligned}$$

where the last inequality is by the following equations:

$$\sum_{i=0}^{\tau-1} \sum_{j=0}^{i-1} i X_j = \sum_{j=0}^{\tau-1} (\sum_{i=m}^{\tau-1} i) X_j \leq \sum_{j=0}^{\tau-1} \tau^2 X_j$$

And

$$\sum_{i=0}^{\tau-1} \sum_{j=0}^{i-1} X_j = \sum_{j=0}^{\tau-1} (\sum_{i=m}^{\tau-1} 1) X_j \leq \sum_{j=0}^{\tau-1} \tau X_j$$

Substituting in (25):

$$\begin{aligned} & \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] \\ & \leq 2\eta_s^2 \frac{P\tau^2 - \tau}{P} \sum_{i=0}^{\tau-1} \mathbb{E}[\|\nabla_{x_s} f_\lambda^t(x_c^t, x_s^{t,i})\|^2] \\ & \quad + 2\eta_s^2\tau \sum_{i=0}^{\tau-1} \left(\frac{4d_s}{P} \mathbb{E}[\|\nabla_{x_s} f_\lambda^t\|^2] + \frac{4L^2 d_s}{P} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] + \frac{2\sigma_L^2 d_s}{P} + \frac{L^2 \lambda^2 d_s^3}{2P} \right) \end{aligned}$$

Further substitute in (30):

$$\begin{aligned}
& \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] \\
& \leq 2\eta_s^2 \frac{P\tau^2 - \tau}{P} \sum_{i=0}^{\tau-1} \left(4\mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 4L^2 \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] + \frac{L^2}{2} \lambda^2 d_s^3 \right) \\
& + 2\eta_s^2 \tau \sum_{i=0}^{\tau-1} \left(\frac{4d_s}{P} \mathbb{E}[\|\nabla_{x_s} f_{\mathbf{x}}^t\|^2] + \frac{4L^2 d_s}{P} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] + \frac{2\sigma_L^2 d_s}{P} + \frac{L^2 \lambda^2 d_s^3}{2P} \right) \\
& \leq \frac{8\eta_s^2 \tau^3 (P + d_s/\tau)}{P} \mathbb{E}[\|\nabla_{x_s} f_{\mathbf{x}}^t\|^2] + \frac{8\eta_s^2 \tau^2 L^2 (P + d_s/\tau)}{P} \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] \\
& + \frac{4\eta_s^2 \tau^2 \sigma_L^2 d_s}{P} + \eta_s^2 \tau^3 L^2 \lambda^2 d_s^3
\end{aligned}$$

Rearranging the terms, we have

$$\begin{aligned}
& \left(1 - \frac{8\eta_s^2 \tau^2 L^2 (P + d_s/\tau)}{P}\right) \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_s^{t,i} - x_s^t\|^2] \\
& \leq \frac{8\eta_s^2 \tau^3 (P + d_s/\tau)}{P} \mathbb{E}[\|\nabla_{x_s} f_{\mathbf{x}}^t\|^2] + \frac{4\eta_s^2 \tau^2 \sigma_L^2 d_s}{P} + \eta_s^2 \tau^3 L^2 \lambda^2 d_s^3
\end{aligned}$$

where we moved the term $\mathbb{E}[\|x_s^{t,i} - x_s^t\|^2]$ to the left in the last inequality. Let $\eta_s \leq \frac{1}{4L\sqrt{\tau^2 + \tau d_s/P}}$, we have the coefficient on the L.H.S larger than $\frac{1}{2}$. Thus, we complete the proof.

D Proof for for MU-SplitFed

D.1 Proof of main theorem

We now prove the main theorem of MU-SplitFed, and defer the important lemmas to Appendix D.2. We re-state the theorem below:

Theorem D.1. *Under Assumption B.1 to B.3, consider a SFL framework with M clients, and let the server iteration number be τ . If the learning rates on client and server satisfy $\eta_c/\tau = \eta_s = \eta \leq \min\{\frac{1}{\sqrt{120L^2(\tau^2 + 2\tau d_s)}}, \frac{M}{12\tau L d_c}\}$, the sequence of iterates generated by MU-Split satisfies:*

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^T \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] & \leq \frac{4}{T\eta_g\eta\tau} \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)] + 24\eta(4\eta\tau L + \eta_g/M)L(\tau + 2d_s)\epsilon^2 \\
& + 16\eta(2\eta\tau L + \eta_g/M)Ld_s\sigma_s^2 + (1/\tau + 8\eta^2\tau L^2 + 2\eta_g\eta/M)\tau L^2\lambda^2 d_s^3 \\
& + \frac{12\eta_g\eta\tau L d_c \epsilon^2}{M} + \frac{4\eta_g\eta\tau L d_c \sigma_c^2}{M} + L^2\lambda^2 d_c^3 \tag{32}
\end{aligned}$$

Similar to the proof of MU-Split, We begin by analyzing the update on client and server side, respectively. By (17), we bound one-round update $\mathcal{K}_1, \mathcal{K}_2$ on the server side, and $\mathcal{K}_3, \mathcal{K}_4$ on the client side.

D.1.1 One-Round Update on Server Side

For \mathcal{K}_1 :

$$\begin{aligned}
& \mathbb{E}[\langle \nabla_{x_s} f(\mathbf{x}^t), x_s^{t+1} - x_s^t \rangle] \\
& = \mathbb{E}[\langle \nabla_{x_s} f(\mathbf{x}^t), -\frac{\eta_g}{M} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \eta_s G_{s,m}^{t,i}(\mathbf{x}_m^{t,i}; \xi_m^t) \rangle]
\end{aligned}$$

$$\begin{aligned}
&= \mathbb{E}[\langle \nabla_{x_s} f(\mathbf{x}^t), -\frac{\eta_g}{M} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \eta_s (\nabla_{x_s} f_{m,\lambda}^{t,i} - \nabla_{x_s} f(\mathbf{x}^t) + \nabla_{x_s} f(\mathbf{x}^t)) \rangle] \\
&= \mathbb{E}[\langle \sqrt{\eta_g \eta_s} \tau \nabla_{x_s} f(\mathbf{x}^t), -\frac{\sqrt{\eta_g \eta_s}}{M \sqrt{\tau}} \sum_{m=1}^M \sum_{i=0}^{\tau-1} (\nabla_{x_s} f_{m,\lambda}^{t,i} - \nabla_{x_s} f_m^t) \rangle] - \eta_g \eta_s \tau \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] \\
&= \frac{\eta_g \eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_s}{2M^2 \tau} \mathbb{E} \left[\left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} (\nabla_{x_s} f_{m,\lambda}^{t,i} - \nabla_{x_s} f_m^t) \right\|^2 \right] \\
&\quad - \frac{\eta_g \eta_s}{2M^2 \tau} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 - \eta_g \eta_s \tau \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] \\
&= -\frac{\eta_g \eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_s}{2M^2 \tau} \mathbb{E} \left[\left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} (\nabla_{x_s} f_{m,\lambda}^{t,i} - \nabla_{x_s} f_m^t + \nabla_{x_s} f_m^t - \nabla_{x_s} f_m^t) \right\|^2 \right] \\
&\quad - \frac{\eta_g \eta_s}{2M^2 \tau} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 \\
&\leq -\frac{\eta_g \eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_s}{M^2 \tau} \mathbb{E} \left[\left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} (\nabla_{x_s} f_{m,\lambda}^{t,i} - \nabla_{x_s} f_m^t) \right\|^2 \right] \\
&\quad + \frac{\eta_g \eta_s}{M^2 \tau} \mathbb{E} \left[\left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} (\nabla_{x_s} f_m^t - \nabla_{x_s} f_m^t) \right\|^2 \right] - \frac{\eta_g \eta_s}{2M^2 \tau} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 \\
&\leq -\frac{\eta_g \eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_s}{M} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \mathbb{E} \left[\left\| (\nabla_{x_s} f_{m,\lambda}^{t,i} - \nabla_{x_s} f_m^t) \right\|^2 \right] \\
&\quad + \frac{\eta_g \eta_s}{M} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \mathbb{E} \left[\left\| (\nabla_{x_s} f_m^t - \nabla_{x_s} f_m^t) \right\|^2 \right] - \frac{\eta_g \eta_s}{2M^2 \tau} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 \\
&\leq -\frac{\eta_g \eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_s}{4} \tau L^2 \lambda^2 d_s^3 + \underbrace{\frac{\eta_g \eta_s L^2}{M} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \mathbb{E} \left[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2 \right]}_{\mathcal{A}_1} \\
&\quad - \frac{\eta_g \eta_s}{2M^2 \tau} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 \\
&\leq -\frac{\eta_g \eta_s \tau}{2} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_s}{4} \tau L^2 \lambda^2 d_s^3 - \frac{\eta_g \eta_s}{2M^2 \tau} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 \\
&\quad + \eta_g \eta_s L^2 \left(24\eta_s^2 (\tau^3 + \tau^2 d_s / P) \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 24\eta_s^2 (\tau^3 + \tau^2 d_s / P) \epsilon^2 + \frac{8\eta_s^2 \tau^2 d_s \sigma_s^2}{P} + 2\eta_s^2 \tau^3 L^2 \lambda^2 d_s^3 \right) \\
&= \left(24\eta_g \eta_s^3 L^2 (\tau^3 + \tau^2 d_s / P) - \frac{\eta_g \eta_s \tau}{2} \right) \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_s}{4} \tau L^2 \lambda^2 d_s^3 + 24\eta_g \eta_s^3 L^2 (\tau^3 + \tau^2 d_s / P) \epsilon^2 \\
&\quad + \frac{8\eta_g \eta_s^3 \tau^2 L^2 d_s \sigma_s^2}{P} + 2\eta_g \eta_s^3 \tau^3 L^4 \lambda^2 d_s^3 - \frac{\eta_g \eta_s}{2M^2 \tau} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2,
\end{aligned}$$

where in the last step we use Lemma D.3 for \mathcal{A}_1 .

For \mathcal{K}_2 :

$$\mathbb{E}[\|x_s^{t+1} - x_s^t\|^2] = \frac{\eta_g^2 \eta_s^2}{M^2} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} G_{s,m}^{t,i}(\mathbf{x}_m^{t,i}; \xi_m^t) \right\|^2$$

$$\leq 2 \frac{\eta_g^2 \eta_s^2}{M^2} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 + 2 \frac{\eta_g^2 \eta_s^2}{M^2} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \mathbb{E} \left\| G_{s,m}^{t,i} - \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2$$

Substituting (34) and (40) in order, we have

$$\begin{aligned} & \frac{\eta_g^2 \eta_s^2}{M^2} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \mathbb{E} \left\| G_{s,m}^{t,i} - \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 \\ & \leq \frac{\eta_g^2 \eta_s^2}{M^2} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \left(\frac{-1}{P} (6\mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2]) + 6\epsilon^2 + 6L^2 \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] + \frac{L^2}{2} \lambda^2 d_s^3 \right) \\ & \quad + \frac{6d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{6L^2 d_s}{P} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] + \frac{6d_s \epsilon^2}{P} + \frac{2d_s \sigma_s^2}{P} + \frac{L^2 \lambda^2 d_s^3}{2P} \Big) \\ & \leq \frac{\eta_g^2 \eta_s^2}{M^2} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \left(\frac{6d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{6L^2 d_s}{P} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] + \frac{6d_s \epsilon^2}{P} + \frac{2d_s \sigma_s^2}{P} \right) \\ & = \frac{\eta_g^2 \eta_s^2}{M} \left(\frac{6d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{6d_s \epsilon^2}{P} + \frac{2d_s \sigma_s^2}{P} \right) \\ & \quad + \frac{6\eta_g^2 \eta_s^2 L^2 d_s}{PM} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] \end{aligned}$$

We then use Lemma D.3, and assume that $\eta_s \leq \frac{\sqrt{P}}{L\sqrt{24\tau d_s}}$. It follows that

$$\begin{aligned} & \frac{\eta_g^2 \eta_s^2}{M^2} \sum_{m=1}^M \sum_{i=0}^{\tau-1} \mathbb{E} \left\| G_{s,m}^{t,i} - \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 \\ & \leq \frac{\eta_g^2 \eta_s^2}{M} \left(\frac{6d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{6d_s \epsilon^2}{P} + \frac{2d_s \sigma_s^2}{P} \right) \\ & \quad + \frac{\eta_g^2}{4\tau M^2} \sum_{m=1}^M (24\eta_s^2 (\tau^3 + \tau^2 d_s / P) \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 24\eta_s^2 (\tau^3 + \tau^2 d_s / P) \epsilon^2 \\ & \quad + \frac{8\eta_s^2 \tau^2 d_s \sigma_s^2}{P} + 2\eta_s^2 \tau^3 L^2 \lambda^2 d_s^3) \\ & \leq \frac{\eta_g^2 \eta_s^2}{M} \left(6(\tau^2 + 2\tau d_s / P) \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 6(\tau^2 + 2\tau d_s / P) \epsilon^2 + \frac{\tau^2 L^2 \lambda^2 d_s^3}{2} + \frac{4\tau d_s \sigma_s^2}{P} \right), \end{aligned}$$

where in the last step we use the fact that $\tau \geq 1$.

D.1.2 One-Round Update on Client Side

For \mathcal{K}_3 :

$$\begin{aligned} & \mathbb{E}[\langle \nabla_{x_c} f(\mathbf{x}^t), x_c^{t+1} - x_c^t \rangle] \\ & = \mathbb{E}[\langle \nabla_{x_c} f(\mathbf{x}^t), -\frac{\eta_g}{M} \sum_{m=1}^M \eta_c G_{c,m}^{t,i}(\mathbf{x}_m^t; \xi_m^t) \rangle] \\ & = \mathbb{E}[\langle \nabla_{x_c} f(\mathbf{x}^t), -\frac{\eta_g}{M} \sum_{m=1}^M \eta_c (\nabla_{x_c} f_{m,\lambda}^t - \nabla_{x_c} f(\mathbf{x}^t) + \nabla_{x_c} f(\mathbf{x}^t)) \rangle] \\ & = \mathbb{E}[\langle \sqrt{\eta_g \eta_c} \nabla_{x_c} f(\mathbf{x}^t), -\frac{\sqrt{\eta_g \eta_c}}{M} \sum_{m=1}^M (\nabla_{x_c} f_{m,\lambda}^t - \nabla_{x_c} f_m^t) \rangle] - \eta_g \eta_c \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] \\ & = \frac{\eta_g \eta_c}{2} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_c}{2M^2} \mathbb{E}[\|\sum_{m=1}^M (\nabla_{x_c} f_{m,\lambda}^t - \nabla_{x_c} f_m^t)\|^2] \end{aligned}$$

$$\begin{aligned}
& -\frac{\eta_g \eta_c}{2M^2} \mathbb{E} \left\| \sum_{m=1}^M \nabla_{x_c} f_{m,\lambda}^t \right\|^2 - \eta_g \eta_c \mathbb{E} [\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] \\
& \leq -\frac{\eta_g \eta_c}{2} \mathbb{E} [\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_c}{4} L^2 \lambda^2 d_c^3 - \frac{\eta_g \eta_c}{2M^2} \mathbb{E} \left\| \sum_{m=1}^M \nabla_{x_c} f_{m,\lambda}^t \right\|^2
\end{aligned}$$

For \mathcal{K}_4 :

$$\begin{aligned}
\mathbb{E} [\|x_c^{t+1} - x_c^t\|^2] &= \frac{\eta_g^2 \eta_c^2}{M^2} \mathbb{E} \left\| \sum_{m=1}^M G_{c,m}^t(\mathbf{x}_m^t; \xi_m^t) \right\|^2 \\
&= \frac{\eta_g^2 \eta_c^2}{M^2} \mathbb{E} \left\| \sum_{m=1}^M \nabla_{x_c} f_{m,\lambda}^t \right\|^2 + \frac{\eta_g^2 \eta_c^2}{M^2} \mathbb{E} \left\| \sum_{m=1}^M (G_{c,m}^t - \nabla_{x_c} f_{m,\lambda}^t) \right\|^2 \\
&\leq \frac{\eta_g^2 \eta_c^2}{M^2} \mathbb{E} \left\| \sum_{m=1}^M \nabla_{x_c} f_{m,\lambda}^t \right\|^2 + \frac{\eta_g^2 \eta_c^2}{M^2} \sum_{m=1}^M \mathbb{E} \|G_{c,m}^t - \nabla_{x_c} f_{m,\lambda}^t\|^2
\end{aligned}$$

Substituting (34) and (40) in order, we have

$$\begin{aligned}
& \frac{\eta_g^2 \eta_c^2}{M^2} \sum_{m=1}^M \mathbb{E} \|G_{c,m}^t - \nabla_{x_c} f_{m,\lambda}^t\|^2 \\
& \leq \frac{\eta_g^2 \eta_c^2}{M^2} \sum_{m=1}^M \left(-\frac{1}{P} \mathbb{E} \|\nabla_{x_c} f_{m,\lambda}^t\|^2 + \frac{6d_c}{P} \mathbb{E} [\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{6d_c \epsilon^2 d_c}{P} + \frac{2d_c \sigma_c^2}{P} + \frac{L^2 \lambda^2 d_c^3}{2P} \right) \\
& \leq \frac{\eta_g^2 \eta_c^2}{M^2} \sum_{m=1}^M \left(-\frac{1}{P} (6\mathbb{E} [\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + 6\epsilon^2 + \frac{L^2}{2} \lambda^2 d_c^3) \right. \\
& \quad \left. + \frac{6d_c}{P} \mathbb{E} [\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{6d_c \epsilon^2}{P} + \frac{2d_c \sigma_c^2}{P} + \frac{L^2 \lambda^2 d_c^3}{2P} \right) \\
& \leq \frac{\eta_g^2 \eta_c^2}{M} \left(\frac{6d_c}{P} \mathbb{E} [\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{6d_c \epsilon^2}{P} + \frac{2d_c \sigma_c^2}{P} \right)
\end{aligned}$$

D.1.3 Server-Client Combination

Putting together:

$$\begin{aligned}
& \mathbb{E} [f(\mathbf{x}^{t+1}) - f(\mathbf{x}^t)] \\
& \leq \left(24\eta_g \eta_s^3 L^2 (\tau^3 + \tau^2 d_s / P) - \frac{\eta_g \eta_s \tau}{2} \right) \mathbb{E} [\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_s}{4} \tau L^2 \lambda^2 d_s^3 + 24\eta_g \eta_s^3 L^2 (\tau^3 + \tau^2 d_s / P) \epsilon^2 \\
& \quad + \frac{8\eta_g \eta_s^3 \tau^2 L^2 d_s \sigma_s^2}{P} + 2\eta_g \eta_s^3 \tau^3 L^4 \lambda^2 d_s^3 - \frac{\eta_g \eta_s}{2M^2 \tau} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 + \frac{\eta_g^2 \eta_s^2 L}{M^2} \mathbb{E} \left\| \sum_{m=1}^M \sum_{i=0}^{\tau-1} \nabla_{x_s} f_{m,\lambda}^{t,i} \right\|^2 \\
& \quad + \frac{\eta_g^2 \eta_s^2 L}{M} \left(6(\tau^2 + 2\tau d_s / P) \mathbb{E} [\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 6(\tau^2 + 2\tau d_s / P) \epsilon^2 + \frac{1}{2} \tau^2 L^2 \lambda^2 d_s^3 + \frac{4\tau d_s \sigma_s^2}{P} \right) \\
& \quad - \frac{\eta_g \eta_c}{2} \mathbb{E} [\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{\eta_g \eta_c}{4} L^2 \lambda^2 d_c^3 - \frac{\eta_g \eta_c}{2M^2} \mathbb{E} \left\| \sum_{m=1}^M \nabla_{x_c} f_{m,\lambda}^t \right\|^2 \\
& \quad + \frac{\eta_g^2 \eta_c^2 L}{2M^2} \mathbb{E} \left\| \sum_{m=1}^M \nabla_{x_c} f_{m,\lambda}^t \right\|^2 + \frac{\eta_g^2 \eta_c^2 L}{M} \left(\frac{3d_c}{P} \mathbb{E} [\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{3d_c \epsilon^2}{P} + \frac{d_c \sigma_c^2}{P} \right) \\
& \leq \left(6\eta_g \eta_s^2 (4\eta_s \tau L + \eta_g / M) L (\tau^2 + 2\tau d_s / P) - \frac{\eta_g \eta_s \tau}{2} \right) \mathbb{E} [\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] \\
& \quad + 6\eta_g \eta_s^2 (4\eta_s \tau L + \eta_g / M) L (\tau^2 + 2\tau d_s / P) \epsilon^2 \\
& \quad + \frac{4\eta_g \eta_s^2 (2\eta_s \tau L + \eta_g / M) \tau L d_s \sigma_s^2}{P} + \frac{\eta_g \eta_s (1/\tau + 8\eta_s^2 \tau L^2 + 2\eta_g \eta_s / M)}{4} \tau^2 L^2 \lambda^2 d_s^3
\end{aligned}$$

$$\begin{aligned}
& + \left(\frac{3\eta_g^2\eta_c^2 L d_c}{MP} - \frac{\eta_g\eta_c}{2} \right) \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{3\eta_g^2\eta_c^2 L d_c \epsilon^2}{MP} + \frac{\eta_g^2\eta_c^2 L d_c \sigma_c^2}{MP} + \frac{\eta_g\eta_c L^2 \lambda^2 d_c^3}{4} \\
\leq & - \frac{\eta_g\eta_s\tau}{4} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 6\eta_g\eta_s^2(4\eta_s\tau L + \eta_g/M)L(\tau^2 + 2\tau d_s/P)\epsilon^2 \\
& + \frac{4\eta_g\eta_s^2(2\eta_s\tau L + \eta_g/M)\tau L d_s \sigma_s^2}{P} + \frac{\eta_g\eta_s(1/\tau + 8\eta_s^2\tau L^2 + 2\eta_g\eta_s/M)}{4} \tau^2 L^2 \lambda^2 d_s^3 \\
& - \frac{\eta_g\eta_c}{4} \mathbb{E}[\|\nabla_{x_c} f(\mathbf{x}^t)\|^2] + \frac{3\eta_g^2\eta_c^2 L d_c \epsilon^2}{MP} + \frac{\eta_g^2\eta_c^2 L d_c \sigma_c^2}{MP} + \frac{\eta_g\eta_c L^2 \lambda^2 d_c^3}{4},
\end{aligned}$$

where we assume

$$\eta_s \leq \frac{1}{\sqrt{120L^2(\tau^2 + 2\tau d_s/P)}}, \quad \eta_c \leq \frac{MP}{12Ld_c}$$

and combine the terms.

To combine the squared norm of the server gradient $\mathbb{E}[\|\nabla_{x_s} F\|^2]$ and client gradient $\mathbb{E}[\|\nabla_{x_c} F\|^2]$, we define the universal step size $\eta := \eta_s$, and let $\eta_c = \eta\tau$. Rearranging the terms, we have

$$\begin{aligned}
\frac{\eta_g\eta\tau}{4} \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] \leq & \mathbb{E}[f(\mathbf{x}^t) - f(\mathbf{x}^{t+1})] + 6\eta_g\eta^2(4\eta\tau L + \eta_g/M)L(\tau^2 + 2\tau d_s/P)\epsilon^2 \\
& + \frac{4\eta_g\eta^2(2\eta\tau L + \eta_g/M)\tau L d_s \sigma_s^2}{P} + \frac{\eta_g\eta(1/\tau + 8\eta^2\tau L^2 + 2\eta_g\eta/M)}{4} \tau^2 L^2 \lambda^2 d_s^3 \\
& + \frac{3\eta_g^2\eta^2\tau^2 L d_c \epsilon^2}{MP} + \frac{\eta_g^2\eta^2\tau^2 L d_c \sigma_c^2}{MP} + \frac{\eta_g\eta\tau L^2 \lambda^2 d_c^3}{4},
\end{aligned}$$

Taking average from $t = 0$ to $T - 1$ at both sides:

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^T \frac{\eta_g\eta\tau}{4} \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] & \leq \frac{1}{T} \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)] + 6\eta_g\eta^2(4\eta\tau L + \eta_g/M)L(\tau^2 + 2\tau d_s/P)\epsilon^2 \\
& + \frac{4\eta_g\eta^2(2\eta\tau L + \eta_g/M)\tau L d_s \sigma_s^2}{P} + \frac{\eta_g\eta(1/\tau + 8\eta^2\tau L^2 + 2\eta_g\eta/M)}{4} \tau^2 L^2 \lambda^2 d_s^3 \\
& + \frac{3\eta_g^2\eta^2\tau^2 L d_c \epsilon^2}{MP} + \frac{\eta_g^2\eta^2\tau^2 L d_c \sigma_c^2}{MP} + \frac{\eta_g\eta\tau L^2 \lambda^2 d_c^3}{4} \\
\frac{1}{T} \sum_{t=0}^T \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] & \leq \frac{4}{T\eta_g\eta\tau} \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)] + 24\eta(4\eta\tau L + \eta_g/M)L(\tau + 2d_s/P)\epsilon^2 \\
& + \frac{16\eta(2\eta\tau L + \eta_g/M)L d_s \sigma_s^2}{P} + (1/\tau + 8\eta^2\tau L^2 + 2\eta_g\eta/M)\tau L^2 \lambda^2 d_s^3 \\
& + \frac{12\eta_g\eta\tau L d_c \epsilon^2}{MP} + \frac{4\eta_g\eta\tau L d_c \sigma_c^2}{MP} + L^2 \lambda^2 d_c^3 \tag{33}
\end{aligned}$$

where in the last step we divided both sides by $\frac{\eta\tau}{4}$. Let $P = 1$, and we complete the proof.

D.1.4 Justification for Corollary 4.4

The optimal convergence rate is achieved by optimizing (33) w.r.t η and η_g , solving which gives $\eta_g = \sqrt{\tau M}$ and $\eta = \frac{1}{\tau L \sqrt{dT}}$. Since d_s, d_c is typically very large, and τ is relatively small, we can assume that $\tau \leq d_s$. Thus, we have

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^T \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] & \leq \frac{4L\sqrt{d}}{\sqrt{\tau TM}} \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)] + \frac{24(4/\sqrt{T} + \sqrt{\tau}/\sqrt{M})(d_s/\sqrt{d})\epsilon^2}{\tau\sqrt{T}} \\
& + \frac{16(2/\sqrt{T} + \sqrt{\tau}/\sqrt{M})(d_s/\sqrt{d})\sigma_s^2}{\tau\sqrt{T}} + (1 + 8/dT + 2\sqrt{\tau}/L\sqrt{dT M})L^2 \lambda^2 d_s^3 \\
& + \frac{12\sqrt{\tau}(d_c/\sqrt{d})\epsilon^2}{\sqrt{TM}} + \frac{4\sqrt{\tau}(d_c/\sqrt{d})\sigma_c^2}{\sqrt{TM}} + L^2 \lambda^2 d_c^3
\end{aligned}$$

Since d, T, M, τ are positive integers and L are typically large, we have that

$$(1 + 8/dT + 2\sqrt{\tau}/L\sqrt{dT M})L^2 \lambda^2 d_s^3 + L^2 \lambda^2 d_c^3 \leq 11L^2 \lambda^2 d^3$$

Let $d/d_c^2 = \tau$, so that $d_c = \sqrt{d/\tau}$ and $d_s = d - \sqrt{d/\tau}$, and further let

$$\lambda^2 = \frac{1}{\tau T d^{5/2} L^2}$$

Finally, we have

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^T \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^t)\|^2] &\leq \frac{4L\sqrt{d}}{\sqrt{\tau TM}} \mathbb{E}[f(\mathbf{x}^0) - f(\mathbf{x}^T)] + \frac{8\sqrt{d}(3\epsilon^2 + 2\sigma_s^2)}{\sqrt{\tau TM}} + \frac{32\sqrt{d}(3\epsilon^2 + \sigma_s^2)}{\tau T} \\ &\quad + \frac{4(3\epsilon^2 + \sigma_c^2)}{\sqrt{TM}} + \frac{6\sqrt{d}}{\tau T} \end{aligned}$$

We can conclude that, the overall convergence rate is $O(\frac{\sqrt{d}}{\sqrt{\tau TM}})$

D.2 Important Lemmas

Lemma D.2 (Bounds on the variance of Zeroth-order Gradient). *Under the same condition as Lemma B.1, and consider the stochastic Zeroth-order Gradient, we can further bound the variance of the local stochastic Zeroth-order Gradient by **global** gradient at the beginning of the local iteration and the local update distance.*

$$\begin{aligned} \mathbb{E}[\|G_s^{t,i}(\mathbf{x}_m^{t,i}; \xi_m^t) - \nabla_{x_s} f_\lambda^t(\mathbf{x}_m^{t,i})\|^2] &\leq \frac{6d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{6L^2 d_s}{P} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] \\ &\quad + \frac{6d_s \epsilon^2}{P} + \frac{2d_s \sigma_s^2}{P} + \frac{L^2 \lambda^2 d_s^3}{2P} - \frac{1}{P} \mathbb{E}[\|\nabla_{x_s} f_\lambda^t(\mathbf{x}_m^{t,i})\|^2] \end{aligned} \quad (34)$$

proof:

First notice that $G_s^{t,i}(\mathbf{x}_m^{t,i}; \xi_m^t) = \frac{1}{P} \sum_{p=1}^P g_{s,p}^{t,i}(\mathbf{x}_m^{t,i}; \xi_m^t)$ and $\mathbb{E}_{u_p, \xi_m^t}[g_{s,p}^{t,i}(\mathbf{x}_m^{t,i}; \xi_m^t)] = \nabla_{x_s} f_\lambda^t(\mathbf{x}_m^{t,i})$. By Lemma B.1, we have

$$\mathbb{E}_u[\|g_{s,p}^{t,i}(\mathbf{x}_m^{t,i}; \xi_m^t)\|^2] \leq 2d_s \cdot \|\nabla_{x_s} F(\mathbf{x}_m^{t,i}; \xi_m^t)\|^2 + \frac{L^2}{2} \lambda^2 d_s^3.$$

Thus we have

$$\begin{aligned} &\mathbb{E}[\|G_s^{t,i}(\mathbf{x}_m^{t,i}; \xi_m^t) - \nabla_{x_s} f_\lambda^t(\mathbf{x}_m^{t,i})\|^2] \\ &= \frac{1}{P^2} \sum_{p=1}^P \mathbb{E}[\|g_{s,p}^{t,i}(\mathbf{x}_m^{t,i}) - \nabla_{x_s} f_\lambda^t(\mathbf{x}_m^{t,i})\|^2] \\ &= \frac{1}{P^2} \sum_{p=1}^P \mathbb{E}[\|g_{s,p}^{t,i}(\mathbf{x}_m^{t,i})\|^2] - \frac{1}{P} \|\nabla_{x_s} f_\lambda^t(\mathbf{x}_m^{t,i})\|^2 \\ &\leq \frac{1}{P} \left[2d_s \mathbb{E}[\|\nabla_{x_s} F(\mathbf{x}_m^{t,i}; \xi_m^t)\|^2] + \frac{L^2}{2} \lambda^2 d_s^3 \right] - \frac{1}{P} \mathbb{E}[\|\nabla_{x_s} f_\lambda^t(\mathbf{x}_m^{t,i})\|^2] \\ &\leq \frac{1}{P} \left[2d_s (\mathbb{E}[\|\nabla_{x_s} f_m^{t,i}\|^2] + \sigma_s^2) + \frac{L^2}{2} \lambda^2 d_s^3 \right] - \frac{1}{P} \mathbb{E}[\|\nabla_{x_s} f_\lambda^t(\mathbf{x}_m^{t,i})\|^2] \\ &= \frac{1}{P} \left[2d_s \mathbb{E}[\|\nabla_{x_s} f_m^{t,i}\|^2] + 2d_s \sigma_s^2 + \frac{L^2}{2} \lambda^2 d_s^3 - \mathbb{E}[\|\nabla_{x_s} f_\lambda^t(\mathbf{x}_m^{t,i})\|^2] \right] \end{aligned} \quad (35)$$

Now we bound the squared norm of the variance:

$$\begin{aligned} \mathbb{E}[\|\nabla_{x_s} f_m^{t,i}\|^2] &= \mathbb{E}[\|\nabla_{x_s} f_m^{t,i} - \nabla_{x_s} f_m^t + \nabla_{x_s} f_m^t - \nabla_{x_s} f(\mathbf{x}^t) + \nabla_{x_s} f(\mathbf{x}^t)\|^2] \\ &\leq 3\mathbb{E}[\|\nabla_{x_s} f_m^{t,i} - \nabla_{x_s} f_m^t\|^2] + 3\mathbb{E}[\|\nabla_{x_s} f_m^t - \nabla_{x_s} f(\mathbf{x}^t)\|^2] + 3\mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] \\ &\leq 3L^2 \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] + 3\mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 3\epsilon^2 \end{aligned} \quad (36)$$

Substituting (36) into (35), and we finish the proof.

Lemma D.3 (Bounds on multiple update steps). *If $\eta_s^t \leq \frac{\sqrt{P}}{\tau L \sqrt{24(P+d_s/\tau)}}$, we have*

$$\begin{aligned} \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] &\leq 24(\eta_s^t)^2(\tau^3 + \tau^2 d_s/P) \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 24(\eta_s^t)^2(\tau^3 + \tau^2 d_s/P)\epsilon^2 \\ &\quad + \frac{8(\eta_s^t)^2 \tau^2 d_s \sigma_s^2}{P} + 2(\eta_s^t)^2 \tau^3 L^2 \lambda^2 d_s^3 \end{aligned}$$

proof:

We first apply the update formula:

$$\sum_{i=0}^{\tau-1} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] = \sum_{i=0}^{\tau-1} (\eta_s^t)^2 \mathbb{E}[\|\sum_{j=0}^{i-1} G_{s,m}^{t,j}(x_c^t, x_s^{t,j})\|^2]$$

By the property martingale difference sequence, we have

$$\begin{aligned} \mathbb{E}[\|\sum_{j=0}^{i-1} G_{s,m}^{t,j}\|^2] &\leq 2\mathbb{E}[\|\sum_{j=0}^{i-1} \nabla_{x_s} f_{m,\lambda}^{t,j}\|^2] + 2\sum_{j=0}^{i-1} \mathbb{E}[\|G_{s,m}^{t,j} - \nabla_{x_s} f_{m,\lambda}^{t,j}\|^2] \\ &\leq 2i \sum_{j=0}^{i-1} \mathbb{E}[\|\nabla_{x_s} f_{m,\lambda}^{t,j}\|^2] + 2\sum_{j=0}^{i-1} \mathbb{E}[\|G_{s,m}^{t,j} - \nabla_{x_s} f_{m,\lambda}^{t,j}\|^2] \end{aligned} \quad (37)$$

We thus have

$$\begin{aligned} \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] &\leq 2\sum_{i=0}^{\tau-1} (\eta_s^t)^2 \left(i \sum_{j=0}^{i-1} \mathbb{E}[\|\nabla_{x_s} f_{m,\lambda}^{t,j}\|^2] + \sum_{j=0}^{i-1} \mathbb{E}[\|G_{s,m}^{t,j} - \nabla_{x_s} f_{m,\lambda}^{t,j}\|^2] \right) \\ &\leq 2(\eta_s^t)^2 \tau^2 \sum_{i=0}^{\tau-1} \mathbb{E}[\|\nabla_{x_s} f_{m,\lambda}^{t,i}\|^2] + (\eta_s^t)^2 \tau \sum_{i=0}^{\tau-1} \mathbb{E}[\|G_{s,m}^{t,i} - \nabla_{x_s} f_{m,\lambda}^{t,i}\|^2], \end{aligned} \quad (38)$$

where the last inequality is by the following equations:

$$\sum_{i=0}^{\tau-1} \sum_{j=0}^{i-1} i X_j = \sum_{j=0}^{\tau-1} \left(\sum_{i=m}^{\tau-1} i \right) X_j \leq \sum_{j=0}^{\tau-1} \tau^2 X_j$$

And

$$\sum_{i=0}^{\tau-1} \sum_{j=0}^{i-1} X_j = \sum_{j=0}^{\tau-1} \left(\sum_{i=m}^{\tau-1} 1 \right) X_j \leq \sum_{j=0}^{\tau-1} \tau X_j$$

Substituting in (34):

$$\begin{aligned} &\sum_{i=0}^{\tau-1} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] \\ &\leq 2(\eta_s^t)^2 \frac{P\tau^2 - \tau}{P} \sum_{i=0}^{\tau-1} \mathbb{E}[\|\nabla_{x_s} f_{m,\lambda}^{t,i}\|^2] \\ &\quad + 2(\eta_s^t)^2 \tau \sum_{i=0}^{\tau-1} \left(\frac{6d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{6L^2 d_s}{P} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] + \frac{6d_s \epsilon^2}{P} + \frac{2d_s \sigma_s^2}{P} + \frac{L^2 \lambda^2 d_s^3}{2P} \right) \end{aligned} \quad (39)$$

From Lemma B.1 we have

$$\mathbb{E}[\|\nabla_{x_s} f_{m,\lambda}^{t,i}\|^2]$$

$$\begin{aligned}
&\leq 2\mathbb{E}[\|\nabla_{x_s} f_m^{t,i}\|^2] + \frac{L^2}{2}\lambda^2 d_s^3 \\
&\leq 2\mathbb{E}[\|\nabla_{x_s} f_m^{t,i} - \nabla_{x_s} f_m^t + \nabla_{x_s} f_m^t - \nabla_{x_s} f(\mathbf{x}^t) + \nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{L^2}{2}\lambda^2 d_s^3 \\
&\leq 6\mathbb{E}[\|\nabla_{x_s} f_m^{t,i} - \nabla_{x_s} f_m^t\|^2] + 6\mathbb{E}[\|\nabla_{x_s} f_m^t - \nabla_{x_s} f(\mathbf{x}^t)\|^2] + 6\mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{L^2}{2}\lambda^2 d_s^3 \\
&\leq 6\mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 6\epsilon^2 + 6L^2\mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] + \frac{L^2}{2}\lambda^2 d_s^3 \tag{40}
\end{aligned}$$

Substitute into (39):

$$\begin{aligned}
&\sum_{i=0}^{\tau-1} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] \\
&\leq 2(\eta_s^t)^2 \frac{P\tau^2 - \tau}{P} \sum_{i=0}^{\tau-1} \left(6\mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 6L^2\mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] + 6\epsilon^2 + \frac{L^2}{2}\lambda^2 d_s^3 \right) \\
&+ 2(\eta_s^t)^2 \tau \sum_{i=0}^{\tau-1} \left(\frac{6d_s}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{6L^2 d_s}{P} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] + \frac{6d_s \epsilon^2}{P} + \frac{2d_s \sigma_s^2}{P} + \frac{L^2 \lambda^2 d_s^3}{2P} \right) \\
&\leq \frac{12(\eta_s^t)^2 \tau^3 (P + d_s/\tau)}{P} \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + \frac{12(\eta_s^t)^2 \tau^2 L^2 (P + d_s/\tau)}{P} \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] \\
&+ \frac{12(\eta_s^t)^2 \tau^3 (P + d_s/\tau) \epsilon^2}{P} + \frac{4(\eta_s^t)^2 \tau^2 d_s \sigma_s^2}{P} + (\eta_s^t)^2 \tau^3 L^2 \lambda^2 d_s^3
\end{aligned}$$

Rearranging the terms, we have

$$\begin{aligned}
&\left(1 - \frac{12(\eta_s^t)^2 \tau^2 L^2 (P + d_s/\tau)}{P}\right) \sum_{i=0}^{\tau-1} \mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2] \\
&\leq 12(\eta_s^t)^2 (\tau^3 + \tau^2 d_s/P) \mathbb{E}[\|\nabla_{x_s} f(\mathbf{x}^t)\|^2] + 12(\eta_s^t)^2 L^2 (\tau^3 + \tau^2 d_s/P) \epsilon^2 \\
&+ \frac{4(\eta_s^t)^2 \tau^2 d_s \sigma_s^2}{P} + (\eta_s^t)^2 \tau^3 L^2 \lambda^2 d_s^3
\end{aligned}$$

where we moved the term $\mathbb{E}[\|x_{s,m}^{t,i} - x_{s,m}^t\|^2]$ to the left in the last inequality. Let $\eta_s^t \leq \frac{\sqrt{P}}{\tau L \sqrt{24(P+d_s/\tau)}}$, we have the coefficient on the L.H.S larger than $\frac{1}{2}$. Thus, we complete the proof.

E Additional Experiments

To investigate the interplay between splitting strategy and unbalanced update frequency τ , we conduct an ablation study examining various combinations of τ values and cutting layers using OPT-1.3B on the SST-2 dataset. To isolate the effects of our core mechanism from confounding factors inherent in federated settings, such as data heterogeneity and client variability, we employ a simplified MU-Split configuration with a single client.

Table 4 shows the total communication round required to attain 85% accuracy across different cut layers and values of τ . For a fixed cut layer (e.g. $L_c = 2$), setting $\tau = 4$ reduces communication rounds by more than half compared to the baseline without unbalanced updates. Crucially, our results reveal a clear trade-off between τ and L_c . When L_c is fixed, increasing τ initially improves convergence, but excessive server updates eventually lead to diminishing or adverse effects. Conversely, when fixing τ and tuning the cut layer, convergence consistently improves as L_c decreases, indicating a deeper server-side model is beneficial for model performance. Moreover, the optimal value of τ shifts higher as L_c moves earlier in the model. These trends confirm our theoretical insight in Section 4: to fully exploit server-side acceleration, the model partition must scale with the number of server iterations.

Table 5 presents the final accuracy after 1,500 training steps under different combinations of split layers and τ . Consistent with the observations in Table 4, for a fixed split layer, increasing τ initially improves the final accuracy but eventually leads to a decline. However, unlike Table 4, when varying the split layer, the highest accuracy is consistently achieved at $\tau = 2$ or $\tau = 3$. This pattern aligns with our theoretical analysis in Section 4: although a larger τ can accelerate convergence, it does not necessarily yield smaller loss value, which is strongly connected to better final accuracy. In practice, selecting appropriate values for τ and the split layer requires balancing multiple factors, including desired training time, target accuracy, and device memory constraints.

Table 4: Ablation study of influence of τ and cutting layer on communication rounds

Split Layer	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$
2	38	17	19	16	18	18
4	-	18	16	22	20	33
8	-	23	22	26	22	32
12	-	22	32	25	29	32
16	-	21	29	28	40	36

Table 5: Ablation study of influence of τ and cutting layer on final accuracy

Split Layer	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$
2	88.75	88.97	90.90	87.95	87.05	88.52
4	-	89.09	89.89	87.05	86.93	89.04
8	-	90.34	90.11	89.50	89.54	88.30
12	-	89.20	89.43	88.41	88.41	88.43
16	-	88.98	88.75	87.95	88.41	87.99

F Choice of Hyperparameters

Table 6: Hyperparameters

PARAMETER	VALUE	EXPLANATION
η_g	0.3	<i>Global aggregation learning rate</i>
η_s	0.01	<i>Server learning rate</i>
η_c	0.005	<i>Client learning rate</i>
λ	0.005	<i>Scale of perturbation for ZOO</i>
B	32	<i>Batch size</i>