# **Fed-OGD**: Mitigating Straggler Effects in Federated Learning via Orthogonal Gradient Descent

Wei Li, Zicheng Shen, Xiulong Liu*, Chuntao Ding, and Jiaxing Shen

***Abstract*—Federated Learning (FL) faces challenges due to straggler clients that impede timely parameter uploads, potentially leading to suboptimal global model performance. Existing approaches using synchronous and asynchronous communication suffer from long waiting times or convergence issues. We propose Fed-OGD, a novel asynchronous FL method addressing the straggler problem through gradient orthogonalization. Our approach innovatively frames the straggler issue using catastrophic forgetting theory, viewing stragglers as instances of the global model "forgetting" to aggregate their parameters. Fed-OGD introduces an Orthogonal Gradient Descent (OGD) technique that caches straggler gradients and orthogonalizes the difference between these and current active client gradients. By projecting active gradients onto straggler orthogonal bases and subtracting the resulting components, we obtain orthogonalized gradients guiding the model towards optimality. We provide theoretical convergence guarantees and demonstrate Fed-OGD's effectiveness through extensive experiments. Our method achieves state-of-the-art performance across multiple datasets among SOTA FL baselines, with notable improvements in non-IID (non-Independent and identically distributed) scenarios: there are few main categories with many samples while other categories hold few samples in a client. Fed-OGD achieves that 16.66% increase in accuracy on CIFAR-10, and significant gains on CIFAR-100 (5.37%), Tiny-ImageNet (38.51%), and AG_NEWS (16.30%).**

***Index Terms*—Federated learning, Straggler issue, Gradient orthogonalization, Catastrophic forgetting**

## I. Introduction

FEDERATED Learning (FL) has emerged as a powerful distributed machine learning paradigm that enables collaborative model training across multiple clients while preserving data privacy [1]. Despite its widespread adoption in privacy-sensitive applications [2]–[5], FL faces significant challenges due to straggler clients—those that fail to upload their parameters to the server for global model aggregation in a timely manner [6]–[8]. This phenomenon can impede global model convergence and degrade overall system performance.

### A. Problem Statement and Motivation

The straggler problem in FL arises from network limitations or device heterogeneity, causing some clients to lag behind others in the training process. In this study, we name the lagged clients as straggler clients, while others are named active clients. The straggler problem is observed in study [9], and it appears in many scenarios in FL. For example, the devices from different participants often exhibit varying computational speeds in IoT [10], and such a device, which holds the slowest speed, usually encumbers the entire FL training as the server must wait for all devices to complete their uploads. Similarly, in wireless edge networks [11], the limited heterogeneous network resources can easily lead to network congestion, which might result in some clients to become the straggler clients because they cannot complete their uploads in a timely manner. These straggler clients significantly increase the overall training time and reduce the efficiency of FL, making them to be a critical issue in practice.

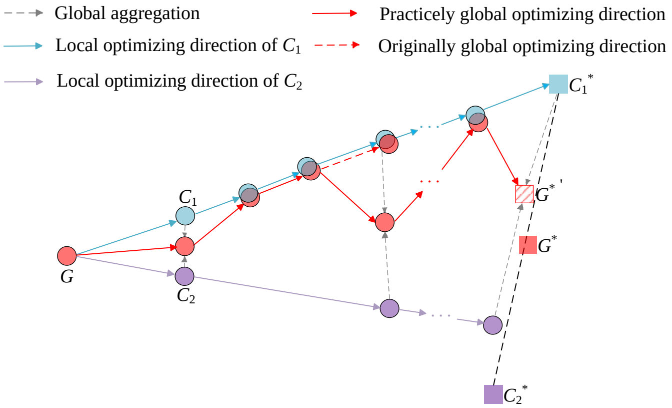

Global aggregation
Practicely global optimizing direction
Local optimizing direction of $C_1$
Originally global optimizing direction
Local optimizing direction of $C_2$
$C_1^*$
$C_1$
$G^{*'}$
$G^*$
$G$
$C_2$
$C_2^*$

Fig. 1: Illustration of optimizing directions from both active client (i.e., $C_1$) and straggler (i.e., $C_2$). $G$ denotes the global model, and its optimizing direction is affected by $C_1$.

As illustrated in Figure 1, when a client ($C_2$) becomes a straggler, the global model ($G$) tends to optimize towards the direction of the active client ($C_1$), potentially reaching a sub-optimal state ($G^{*'}$) instead of the true optimum ($G^*$). This occurs because $C_2$ uploads its parameters to the server less frequently, causing the parameter aggregation of $G$ to be primarily influenced by $C_1$.

Traditional approaches to mitigate the straggler issue, such as increasing bandwidth or upgrading hardware, are limited by diminishing returns [12]. An intuitive method is to abandon the stragglers in FL. Obviously, this method might result in convergence difficulty for the global model when suffering from stragglers. Straggler clients might hold the unique data that are crucial for the global model's convergence, and discarding them could cause the global model to deviate from its optimal status. Recent FL research has explored

Wei Li and Zicheng Shen are with the School of Artificial Intelligence and Computer Science & Engineering Research Center of Intelligent Technology for Healthcare, Ministry of Education & Jiangsu Key Laboratory of Media Design and Software Technology, Jiangnan University, Jiangsu, P. R. China. E-mail: *cs_weili@jiangnan.edu.cn*; *zicheng_shen@stu.jiangnan.edu.cn*.

Xiulong Liu is with College of Intelligence and Computing, Tianjin University, Tianjin, P. R. China. Corresponding author. E-mail: *xiulong_liu@tju.edu.cn*

Chuntao Ding. School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. E-mail: *chtding@bjtu.edu.cn*

Jiaxing Shen. School of Data Science, Lingnan University, Hong Kong, P. R. China. E-mail: *jiaxingshen@ln.edu.hk*

synchronous and asynchronous communication strategies [1], [13]–[15]. However, these methods have inherent limitations: synchronous approaches are hard to avoid waiting for straggler updates, while asynchronous methods may struggle with convergence, especially in non-IID data scenarios [16].

### B. Limitations of Prior Art

Synchronous communication methods ensure consistent client participation in averaging aggregation for the global model but at the cost of increased training time. Studies have attempted to optimize this process by selecting specific clients (e.g. FedProx [17], FedCS [18]) or dividing clients into sets based on response time (e.g. TiFL [19], FedHiSyn [20]). However, these approaches still incur additional training time.

Asynchronous approaches offer faster and more flexible training strategies by uploading parameters (e.g. FedAsync [13]) or gradients (e.g. MIFA [21]) from only the active clients when stragglers arise. Yet, such approaches may bias the global model towards the optimal status of active clients [17], hindering convergence. Attempts to store straggler information at the server [21], [22] have not fully addressed the convergence issue, as they merely cache straggler parameters or gradients without adequately addressing the global model's optimization direction.

### C. Our Approach

We propose **Fed-OGD**, a novel asynchronous FL algorithm that addresses the straggler problem through gradient orthogonalization. Our approach is inspired by the observation that the straggler issue shares similarities with Catastrophic Forgetting (CF) [23], [24] in neural networks, where the global model "forgets" to aggregate parameters from stragglers. We innovatively utilize CF theory to explain and theoretically prove the global model convergence difficulty.

Specifically, we view the latest cached gradients of stragglers as orthogonal bases for the active clients and vice versa. We employ Orthogonal Gradient Descent (OGD) [25] to reconcile the optimization direction of the global model by reducing the difference between the orthogonal bases of stragglers and the current gradients of active clients. By caching straggler gradients on the server, we project the current gradients of active clients onto the orthogonal bases of stragglers to obtain projected components, which implicitly indicate the difference between the optimizing directions of active clients and stragglers.

We perform OGD when the values of the projected components are negative, given that their direction opposes that of the orthogonal bases. The orthogonalized gradients, obtained by subtracting these projected components from the gradients of active clients, are then used to update the parameters of active clients through back-propagation. This process is applied symmetrically to stragglers.

Unlike previous orthogonalization methods that can affect the optimizing direction of the model using gradient rotation [26], [27], **Fed-OGD** avoids the limitations of Householder and Givens transformations, providing a more effective solution to the straggler problem in FL.

### D. Contributions and Advantages

Our key contributions are as follows:

- We propose a novel theoretical framework for explaining global model convergence difficulties in FL using Catastrophic Forgetting theory.
- We introduce **Fed-OGD**, a new FL algorithm that employs gradient projection and orthogonalization to guide the global model towards its optimal status and ensure convergence.
- We present comprehensive experiments on three image classification datasets, demonstrating the superiority of **Fed-OGD** over state-of-the-art (SOTA) baseline models and validating the significance of its components through ablation studies.

The remainder of this paper is structured as follows: Section II reviews FL and its straggler mitigation strategies. Section III details our **Fed-OGD** approach. Section IV presents experimental results and analysis. We conclude in Section V with a summary of our findings and directions for future work.

## II. Related Work

This section provides a comprehensive overview of Federated Learning (FL) and discusses existing studies addressing the straggler issue in FL, categorized by their communication strategies. We also explore the underlying challenges and limitations of current approaches, setting the stage for our proposed method.

### A. Federated Learning Objective

Federated Learning (FL) is a distributed machine learning paradigm that aims to collaboratively train a shared, generalized global model through local training with averaging aggregation on a central server. This approach allows for privacy-preserving model training across multiple decentralized edge devices or servers holding local data samples, without exchanging them [1]. The objective function of vanilla FL can be formulated as: $\min_w F(w) := \min_w \mathbb{E}_{k=1}^{K}[\frac{1}{K}\mathcal{L}_k(w)] = \min_w \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_k(w)$ where $\mathcal{L}_k(w)$ represents the objective function of the $k$-th local model, $w$ denotes its parameters, $F(w)$ is the average loss across all clients, and $K$ indicates the total number of local devices.

Following the principle of Empirical Risk Minimization [28], FL typically employs stochastic gradient descent (SGD) or its variants to find the optimal $w^*$ that minimizes $F(w)$. The training process in FL generally follows these steps:

The server initializes the global model parameters. A subset of clients is selected to participate in the current round. The selected clients download the global model. Clients perform local training on their private data. Clients upload their updated model parameters to the server. The server aggregates the received updates to improve the global model. This process is repeated for multiple rounds until convergence or a predefined stopping criterion is met. However, in practice, some clients fail to upload their parameters to the server due to bandwidth limitations, device issues, or other constraints, leading to the straggler problem. This issue can significantly impact the efficiency and effectiveness of the FL process.

### *B. Synchronous Communication Approaches*

Synchronous communication strategies require the server to receive parameters from all clients at each epoch, necessitating additional waiting time for stragglers. Several approaches have been proposed to address this issue:

*1) Proximal and Selection-based Methods*

FedProx [17] introduces a proximal term to the traditional FL loss function, encouraging local models to stay close to the global model. The modified objective function can be expressed: $\min_w \frac{1}{K}\sum_{k=1}^{K}\left(\mathcal{L}_k(w)+\frac{\mu}{2}|w_k-w|^2\right)$ where $w_k$ is the local model of $k$-th client.and where $w_t$ indicates the global model at round $t$, and $\mu$ denotes a hyperparameter controlling the strength of the proximal term. While this approach helps mitigate the negative impact of non-IID data and system heterogeneity, it does not directly address the straggler issue and increases training cost.

FedCS [18] attempts to improve upon FedProx by selecting active clients with the smallest estimated training time. This method employs a two-stage protocol:

Resource Request: Clients report their available resources to the server. Client Selection: The server selects a subset of clients based on their reported resources and estimated completion time. While this reduces waiting time, it may bias the global model towards active clients, potentially compromising the model's generalization ability.

Recent work [15] proposes a Value of Information (VOI) metric for client selection, considering factors such as loss values, dataset size, model staleness, and upload intervals. However, this approach still faces challenges in reducing training time and mitigating bias towards active clients.

*2) Hierarchical Methods*

Tier-based Federated Learning (TiFL) [19] and FedHiSyn [20] employ hierarchical approaches, dividing clients into sets based on response time or computational performance. TiFL groups clients into tiers and performs synchronous updates within each tier, while allowing asynchronous updates between tiers. The update rule for TiFL can be expressed as: $w_{t+1}=w_t+\eta\sum_{i=1}^{T}\alpha_i\sum_{k\in S_i}\frac{|D_k|}{\sum_{j\in S_i}|D_j|}(w_k-w_t)$ where $T$ represents the number of tiers, $S_i$ denotes the set of clients in tier $i$, and $\alpha_i$ is the weight assigned to tier $i$.

FedHiSyn introduces a hierarchical structure with multiple levels of aggregation to reduce communication overhead and improve scalability. While these methods aim to reduce waiting time for stragglers, they face practical challenges in set formation and may increase communication costs or degenerate to FedAvg performance in certain scenarios.

### *C. Asynchronous Communication Approaches*

Asynchronous communication strategies avoid waiting for stragglers but face challenges in ensuring global model convergence due to the potential staleness of updates.

*1) Parameter Update Strategies*

FedAsync [13] allows clients to upload parameters at any time, with the server aggregating upon receipt. The update rule for FedAsync can be expressed as: $w_{t+1}=w_t+\eta_t\alpha(t,\tau)(w_k-w_t)$ where $\eta_t$ indicates the learning rate, $\alpha(t,\tau)$ is a staleness-based weighting function, and $\tau$ denotes the staleness of the update. However, this approach struggles with stale parameters from stragglers, which can negatively impact model convergence.

Temporally Weighted Asynchronous Federated Learning (ASTW) [29] attempts to address this by adjusting weights for stale parameters: $w_{t+1}=w_t+\eta_t\frac{e^{-\beta\tau}}{\sum_{j=1}^{K}e^{-\beta\tau_j}}(w_k-w_t)$ where $\beta$ is a hyperparameter controlling the decay rate of stale updates. While this approach mitigates the impact of stale updates, it may still bias the model towards active clients.

*2) Server-side Caching*

Memory-augmented Impatient Federated Averaging (MIFA) [21] and similar approaches like TEA-fed [14] and CA$^2$FL [30] store straggler gradients on the server. MIFA introduces a server-side memory module to cache and reuse stale gradients: $w_{t+1}=w_t-\frac{\eta_t}{|S_t|}\sum_{k\in S_t}g_k+\frac{\eta_t}{|M_t|}\sum_{m\in M_t}g_m$ where $S_t$ indicates the set of clients participating in round $t$, $M_t$ is the set of cached gradients, and $g_k$ and $g_m$ are the gradients from active clients and cached memory, respectively. However, these methods still struggle with optimizing the global model direction effectively, especially in highly heterogeneous settings.

*3) Advanced Optimization Techniques*

Gradient-Memory-based Accelerated Federated Learning (GradMA) [31] formulates the optimization as a Quadratic Programming (QP) problem: $\min_\alpha \frac{1}{2}\alpha^TQ\alpha+c^T\alpha \quad \text{s.t.} \quad \sum_{i=1}^{N}\alpha_i=1,\alpha_i\geq 0$ where $Q$ is a matrix of inner products between gradients, and $c$ is a vector of inner products between gradients and the current model update. This approach aims to find optimal weights for combining gradients from different clients and time steps.

Hybrid Federated Learning (HFL) [32] employs Taylor expansion to approximate straggler gradients: $g_k(w_t)\approx g_k(w_{t-\tau})+H_k(w_{t-\tau})(w_t-w_{t-\tau})$ where $H_k$ means an approximation of the Hessian matrix. While these approaches introduce sophisticated optimization techniques to address the straggler problem, they face computational challenges in practice, especially for large-scale models and datasets.

### *D. Hybrid Approaches*

Semi-Asynchronous Federated Averaging (SAFA) [22] introduces a hybrid approach, waiting for a subset of stragglers and randomly selecting clients for aggregation. The update rule for SAFA can be expressed as: $w_{t+1}=w_t+\eta_t\frac{1}{|S_t|}\sum_{k\in S_t}(w_k-w_t)$ where $S_t$ indicates a subset of clients selected for aggregation in round $t$. While this method attempts to balance synchronous and asynchronous benefits, it still incurs waiting time and may lead to suboptimal global model convergence due to the random selection of clients.

### *E. Research Gap and Motivation*

Despite the diverse approaches proposed in these literature, existing methods for addressing the straggler problem in FL face significant challenges. Synchronous methods often incur substantial waiting times or introduce biases towards active clients, compromising the efficiency and fairness of the FL process. Asynchronous methods struggle with convergence due to stale parameters and suboptimal aggregation strategies, potentially leading to unstable or inferior global models. Hybrid approaches, while promising, still face trade-offs between waiting time and model quality, and may not fully leverage the information from all clients. Most existing methods do not adequately address the underlying causes of stragglers, such as system heterogeneity and non-IID data distributions. There is a lack of theoretical guarantees for many proposed approaches, particularly in terms of convergence rates and model quality in the presence of stragglers. These limitations motivate our research to develop a novel approach that can address the straggler problem while maintaining model efficiency in FL. Our proposed method aims to tackle these challenges by introducing a new paradigm that combines adaptive client selection, intelligent gradient aggregation, and theoretical guarantees for convergence and performance.

## III. Fed-OGD

### A. Convergence Difficulty Explanation with CF

The definition of Catastrophic Forgetting (CF) in continual learning is the model "forgetting" knowledge from old tasks when learning the knowledge of new tasks. In FL, since the global model aggregation appears at each epoch. When the stragglers miss to aggregate the parameters of the global model that can be viewed as the global model "forgetting" the parameters from stragglers when facing the parameters from active clients. In such a scenario, the global model is significantly affected by the active clients, thus causing a convergence difficulty. In this way, we explain the difficulty of global model convergence with CF.

Assuming there are two types of clients (i.e. active clients and stragglers). Let $f^r$ be the status of the global model at $r$-th epoch, and let $\Delta f_a^{r+t}$ be the sum of gradients from active clients at $(r+t)$-th epoch (i.e., $\Delta f_a^{r+t} = f_a^{r+t} - f_a^{r+t-1}$). $\Delta f_s^{r+1}$ is the sum of gradients from stragglers. The objective function of global model is shown as follows:

$$f^{r+T} = f^r + \Delta f_s^{r+1} + \sum_{t=2}^{T} \Delta f_a^{r+t} = f^{r+1} + \sum_{t=2}^{T} \Delta f_a^{r+t} \quad (1)$$

where $T$ denotes the interval epoch. It means that the stragglers only uploads their gradients to server every $T$ epochs.

In continual learning, let $c_0$ be the initial status of the model and also assuming there are $n$ tasks. Let $c_1$ be the first status of model after training it on task 1, $c_2$ be the second status of model after training it on task 2, ..., and $c_n$ be the last status of this model after training it on task $n$. Besides, let $\Delta c_n$ be the sum of gradients calculated on task $i$ (i.e., $\Delta c_i = c_i - c_{i-1}$), the process of the model updating in continual learning is defined as follows:

$$c_n = c_0 + \sum_{i=1}^{n} \Delta c_i = c_1 + \sum_{i=2}^{n} \Delta c_i \quad (2)$$

To explain the convergence difficulty with CF, here we assume $n = 2$, i.e., one active client and one straggler in FL and 2 tasks in continual learning. With this, we can regard the process of model updating in continual learning as the model moving away from its optimal status in task 1 to approach to that in task 2. In other words, this model is heavily affected by the task 2 rather than task 1, because new task (i.e., task 2) could cause it "forgetting" the knowledge of old task (i.e., task 1). Under such a scenario, there is an optimizing direction difference between task 1 and task 2. For the straggler issue in FL, the global model "forgets" the knowledge from straggler, so its optimization is mainly affected by the updates from active client. In other words, there is also a difference between the straggler and the active client. Besides, since the straggler does not participate in the aggregation of the global model, it is difficult to get converged. Considering the similar "forgetting" scenario and the optimizing direction difference dilemma in both FL and continual learning, we innovatively explain the global model convergence difficulty with CF in this paper. We now state this properly in the following.

**Proposition 1.** Let $T$ be the interval epoch and $f$ denotes the global model status. $f_s^*$ indicates the optimal status of straggler. If there is a difference between the optimizing direction of active client and straggler at $r$-th epoch in FL, we can explain Eq. (1) with Eq. (2) (i.e., Eq. (1) $\approx$ Eq. (2)).

**Proof.** In continual learning, the status of model (i.e., $c_1$) becomes optimal (i.e., $c_1^*$) after training this model in task 1. Its status is far away from $c_1^*$ and approaches $c_2^*$ when facing the new task 2. Assuming the status of model is $c_2$ at a certain epoch in task 2, we can observe that the distance between $c_2$ and $c_1^*$ is less than that between $c_1$ and $c_1^*$, i.e., $\|c_2 - c_1^*\| < \|c_1 - c_1^*\|$. Such a scenario is also shown in straggler issue, because the optimal status of global model approaches that of active client. Assuming the global model receives parameters from both straggler and active client at $r$ epoch, and only receives parameters from active client from $r+1$ to $r+T$ epochs. Obviously, the distance between the status of global model and optimal status of straggler (i.e., $f_s^*$) at $r+1$ epoch is less than that at $r+T$ epoch, i.e., $\|f^{r+1} - f_s^*\| < \|f^{r+T} - f_s^*\|$. Here, we expand $\|f^{r+T} - f_s^*\|$:

$$\begin{aligned}
\left\|f^{r+T} - f_s^*\right\|^2 &= \left\|f^{r+1} - f^{r+1} + f^{r+T} - f_s^*\right\|^2 \\
\Rightarrow \left\|f^{r+T} - f_s^*\right\|^2 - \left\|f^{r+1} - f_s^*\right\|^2 &= \left\|f^{r+1} - f^{r+T}\right\|^2 \\
&\quad + 2cos(f^{r+1} - f^{r+T}, f^{r+1} - f_s^*) \\
&\quad \cdot \left\|f^{r+1} - f^{r+T}\right\| \cdot \left\|f^{r+1} - f_s^*\right\| \\
\Rightarrow \frac{\left\|f^{r+T} - f_s^*\right\|^2 - \left\|f^{r+1} - f_s^*\right\|^2}{\left\|f^{r+1} - f^{r+T}\right\| \cdot \left\|f^{r+1} - f_s^*\right\|} &= \frac{\left\|f^{r+1} - f^{r+T}\right\|}{\left\|f^{r+1} - f_s^*\right\|} \\
&\quad + 2cos(f^{r+1} - f^{r+T}, f^{r+1} - f_s^*)
\end{aligned}$$

Since there is a difference between the optimizing directions of active client and straggler at $r$-th epoch, we have $cos(f^r - f_a^*, f^r - f_s^*) < 0$. According to Eq. (1), it is obvious that the global model updates along the optimizing direction of straggler at $(r+1)$-th epoch and updates along the optimizing direction of active client from epoch $r+2$ to epoch $r+T$. In this way, we obtain $\cos(f^r - f_a^*, f^r - f_s^*) \approx \cos(f^{r+1} - f^{r+T}, f^{r+1} - f_s^*) < 0$. With this, we deduce $\cos(f^{r+1} - f^{r+T}, f^{r+1} - f_s^*)$ in following:

$$\cos(f^{r+1} - f^{r+T}, f^{r+1} - f_s^*) < 0$$

$$\begin{aligned}
&\Rightarrow \cos(f^{r+1} - f^{r+T}, f^{r+1} - f_s^*) < 0 < \frac{\|f^{r+1} - f^{r+T}\|}{\|f^{r+1} - f_s^*\|} \\
&\qquad < -\cos(f^{r+1} - f^{r+T}, f^{r+1} - f_s^*) \\
&\Rightarrow \frac{\|f^{r+1} - f^{r+T}\|}{\|f^{r+1} - f_s^*\|} + 2\cos(f^{r+1} - f^{r+T}, f^{r+1} - f_s^*) \\
&\qquad < \cos(f^{r+1} - f^{r+T}, f^{r+1} - f_s^*) < 0 \\
&\qquad \Rightarrow \|f^{r+T} - f_s^*\| - \|f^{r+1} - f_s^*\| < 0
\end{aligned}$$

Here, $r$ satisfies $\|f^{r+T} - f_s^*\| < \|f^{r+1} - f_s^*\|$, which is similar to $\|c_2 - c_1^*\| < \|c_1 - c_1^*\|$. In this way, $f^{r+1}$, $f^{r+T}$, and $f_s^*$ can be explained as $c_1$, $c_2$, and $c_1^*$. $f^{r+T} = f^{r+1} + \sum_{t=2}^{T} \Delta f_a^{r+t}$ (i.e.,, Eq. (1)) can also be explained by $c_2 = c_1 + \sum_{t=2}^{T} \Delta f_a^{r+t}$. (i.e., Eq. (2) Considering that both $\sum_{t=2}^{T} \Delta f_a^{r+t}$ and $\Delta c_2$ represent updates on a task, $f^{r+1} + \sum_{t=2}^{T} \Delta f_a^{r+t} \approx c_2 = c_1 + \Delta c_2$, i.e., Eq. (1) $\approx$ Eq. (2).

There are many methods that are used to address CF (e.g. distribution matching [33], data replay [34], and OGD [25]). However, both distribution matching and data replay have a potentially privacy issue [35] and this is against the principle of FL, while OGD does not have such an issue. Hence, we utilize OGD to address the convergence difficulty. Since directly applying the traditional OGD to FL might cause it to be difficult to quantitatively measure the difference between the optimizing directions of both straggler and active client, we present a new OGD to measure this difference, thereby addressing the global model convergence difficulty and thus improving the global model performance. Since FL includes global model updating and local model updating, we discuss them in the following, respectively.

**Algorithm 1: Fed-OGD**

```
Input: number of local step $I$, global model $w^0$.
Output: Prediction results from the global network.
for each epoch $t$ = 0,1,$\cdots$, T-1 do
    for $i \in S_1$ parallel do
        $\Delta w_i^t \leftarrow$ ClientUpdate($i$, $w^t$, $b_2$);
        update $\Delta w_i^{t-1}$ in $M_2$ by $\Delta w_i^t$;
    end
    for $j \in S_2$ ready to training next round do
        $\Delta w_j^{t+1} \leftarrow$ ClientUpdate($j$, $w^t$, $b_1$);
        update $\Delta w_j^t$ in $M_2$ by $\Delta w_j^{t+1}$
    end
    $b_1 \leftarrow \frac{1}{|M_1|} \sum_{k=1}^{|M_1|} \Delta w_k$, $b_2 \leftarrow \frac{1}{|M_2|} \sum_{k=1}^{|M_2|} \Delta w_k$;
    $\Delta w^t = b_1 + b_2$, $w^{t+1} \leftarrow w^t - \eta_g \Delta w^t$;
end
Function ClientUpdate($k$, $w^t$, $b$):
    $w_k^{t,0} \leftarrow w^t$ ;
    for each local step $\tau$ = 0,1,$\cdots$, I-1 do
        $g_k^\tau \leftarrow \Delta f(w_k^{t,\tau})$, $proj(g_k^\tau, b) = \frac{g_k^\tau \cdot b}{\|b\|^2} \cdot b$;
        if $proj(g_k^\tau, b) < 0$ then
            $\tilde{g} \leftarrow g_k^\tau - proj(g_k^\tau, b)$;
        else
            $\tilde{g} \leftarrow g_k^\tau$;
        end
        Update local model $w_k^{t,\tau+1} \leftarrow w_k^{t,\tau} - \eta_l \tilde{g}$;
    end
    Return $\frac{w_k^{t,\tau+1} - w^t}{\eta_l}$;
Function End
```

### B. Global Model Updating

In our **Fed-OGD**, the server constructs a key-value dictionary, $M$, and its length is equivalent to the number of all clients. It caches the latest gradients from all clients. $M$ can be divided into $M_1$ and $M_2$, respectively. $M_1$ is utilized to cache the gradients from straggler, and $M_2$ caches the gradients from active clients. The key in the dictionary contains the information about whether the current gradients belong to $M_1$ or $M_2$, while the value holds the latest uploaded gradients from all clients. If a client becomes an active client, its key will indicate that its gradient belongs to $M_2$, and the straggler's gradient cached in value will be replaced with the new gradient from the active client, because active client uploads its gradients at each epoch. In contrast, if a client becomes a straggler client, its key will indicate that its gradient belongs to $M_1$, and the gradient cached in value will remain unchanged until it uploads. The gradients cached in server are represented as $\Delta w_k^t = \frac{w_k^{t,I} - w^t}{\eta_l}$ where $t$ represents the $t$-th epoch and $I$ denotes the number of iterations. $\eta_l$ indicates the learning rate of client (i.e., local learning rate). Moreover, $w^t$ indicates the parameters of the global model at $t$-th epoch. $w_k^{t,I}$ means the parameters of $k$-th local model, no matter it is active or straggler, at $t$-th epoch. $\Delta w_k^t$ denotes the sum of gradients from the $k$-th client at the $t$-th epoch.

Then, the server calculates two orthogonal bases before aggregation. The calculated value represents the latest optimizing direction of the global model, which is determined by active clients and stragglers rather than active clients only, and they are denoted as $b_1$ and $b_2$, respectively. $b_1$ ($b_2$) is obtained by averaging the gradients in $M_1$ ($M_2$) and is then transmitted to all straggler clients (active clients) who participate in the subsequent training. The calculation of the orthogonal basis is shown as follows:

$$b_i = \frac{1}{|M_i|} \sum_{k=1}^{|M_i|} \Delta w_k \quad i \in \{1, 2\} \tag{3}$$

where both orthogonal bases, $b_1$ and $b_2$, are initialized to 0. Then, the orthogonal base of stragglers is transmitted to the active clients, which provides the optimizing direction of stragglers for active clients to perform OGD. Similarly, the orthogonal basis of active clients is transmitted to stragglers.

Last, the server calculates the updates from the global model by formatting $\Delta w^t = \sum_{k=1}^{K} \Delta w_k$ where $\Delta w_k$ denotes the latest gradients remained in both $M_1$ and $M_2$ for the global model aggregation. According to Eq. (3), the global model updates can be transformed to $\Delta w^t = b_1 + b_2$. After that, we utilize $w^{t+1} = w^t - \eta_g \Delta w^t$ to update the parameters of the global model where $\eta_g$ indicates the global learning rate. In this way, the global model finishes its update.

Our **Fed-OGD** handles dynamic straggler behavior or varying participation rates by transferring the caching location of the gradient in server. Assuming there are 30 active clients and 20 stragglers at $t$-th epoch, and there are 10 active clients and 40 stragglers at $(t+1)$-th round, which means that there are 20 active clients becoming straggler ones. Here, we assume that $M_2^t = \{g_1, ..., g_{30}\}$ caches the gradients from the 30 active clients at $t$-th epoch, and $M_1^t = \{g_{31}, ..., g_{50}\}$ caches the gradients of these 20 stragglers at $t$-th epoch in server, where $g_i$ denotes the gradient of $i$-th client. Assuming the gradients of clients that become straggler clients from active clients are denoted as $\{g_1, ..., g_{20}\}$, these gradients, cached in $M_2^t$ are directly transferred to $M_1^{t+1}$, (i.e. $M_2^{t+1} = M_2^t - \{g_1, ..., g_{20}\} = \{g_{21}, ..., g_{30}\}$ and $M_1^{t+1} = M_1^t + \{g_1, ..., g_{20}\} = \{g_1, ..., g_{20}, g_{31}, ..., g_{50}\}$). The server utilizes the gradient of $M_1^{t+1}$ and $M_2^{t+1}$ to compute the orthogonal bases, $b_1$ and $b_2$, at $(t+1)$-th epoch.

Additionally, since the **Fed-OGD** transmits and caches the gradients is in an unencrypted manner in server, attackers may infer the input information of clients or even reconstruct the original data by analyzing the cached gradients. To avoid this, clients can keep their gradients $\Delta w_k^{t-1}$ from the last training to avoid the privacy risks rather than having the latest gradients in the server cache. Specifically, clients upload the difference $\Delta b_k = \Delta w_k^t - \Delta w_k^{t-1}$ to the server after the local training for orthogonal calculation, with $b_i = b_i + \Delta b_k$. Obviously, this does not give the impact on calculating the orthogonal base and the global updating. Besides, the uploading differences can avoid the gradient inference attack.

### C. Updating of Local Models

Although caching the gradients of stragglers helps alleviate the "forgetting" scenario, the global model still tends to converge towards the optimal status of the active clients, because the averaging aggregation of the global model is mainly affected by the active clients during training and the cached gradients are stale. To address this issue, we present a new OGD strategy, which includes two steps: 1) the active clients project their current gradients on the orthogonal bases of stragglers to obtain the projected components and then measure the difference between current gradients and orthogonal bases based on the projected components; 2) active clients would orthogonalize gradients by subtracting the projected component from current gradients of these active clients to obtain the orthogonalized gradients when suffering from a difference between current gradients and orthogonal bases. After that, we utilize those orthogonalized gradients to update the parameters of active clients with back-propagation. This OGD operation is the same for straggler clients.

In our study, a gradient is represented by a vector [36], and we view the latest cached gradients of stragglers as orthogonal bases for active clients and also view the last gradients of active clients as orthogonal bases for stragglers. In this way, the projection indicates such a component that a vector is projected on the direction of another vector, and we calculate the projected component by following steps. Assuming that the gradient of $k$-th client at $\tau$-th iteration is $g_k^\tau$ and the orthogonal base is $b$. First, we flatten the tensor of gradient and orthogonal base into a one-dimensional vector. Then, we calculate the square of the norm of the orthogonal base (i.e., $\|b\|^2$), and perform the inner product between $g_k^\tau$ and $b$ (i.e., $\langle g_k^\tau, b\rangle$). Third, we calculate the quotient of $\langle g_k^\tau, b\rangle$ and $\|b\|^2$, with $\frac{\langle g_k^\tau, b\rangle}{\|b\|^2}$. Last, we multiply the result of the quotient by $b$ to obtain the projected component, $proj(g_k^\tau, b)$. The formula can be written as $proj(g_k^\tau, b) = \frac{g_k^\tau \cdot b}{\|b\|^2} \cdot b$. If $\langle g_k^\tau, b\rangle < 0$, the two directions are different, so the difference appears; while $\langle g_k^\tau, b\rangle > 0$, the two directions are parallel to each other, and there is no difference.

After that, we orthogonalize the gradients of active clients by subtracting the projected components from the current gradients of the same clients when $proj(g_k^\tau, b_2) < 0$, that is $\tilde{g} = g_k^\tau - proj(g_k^\tau, b_2)$. With this, the orthogonalized gradients, $\tilde{g}$, are orthogonal to $b_2$. $\tilde{g} = g$ when $proj(g_k^\tau, b_2) \geq 0$. In such a scenario, we do not perform OGD. Such an OGD strategy reduces the difference between optimizing directions of both active clients and stragglers by removing the projected components in different directions from the current gradients of active clients or retaining the projected components in parallel direction from the current gradients of active clients. This can be formulized as follows:

$$\tilde{g} = \begin{cases} g_k^\tau - \frac{\langle g_k^\tau, b_2\rangle}{\|b_2\|^2} \cdot b_2 & if \quad \langle g_k^\tau, b_2\rangle < 0 \\ g_k^\tau & if \quad \langle g_k^\tau, b_2\rangle \geq 0 \end{cases} \tag{4}$$

After obtaining gradient orthogonalization, we utilize $w_k^{t,\tau+1} = w_k^{t,\tau} - \eta_l \tilde{g}$ to update the parameters of the global model where $\eta_g$ denotes the learning rate of the global model. Note that such an orthogonalization operation is also the same as the stragglers. For the stragglers, they receive the orthogonal base, $b_1$, which is the averaging gradients of the active clients. This guarantees that the optimizing directions of active clients are not deviating far away from their optimal status, while guiding the optimizing direction of the global model to its optimal status rather than the active clients. The pseudo-codes of **Fed-OGD** is presented in **Algorithm 1**, and its convergence is proven in the following.

The orthogonalization subtracts a projected component from the gradient, which results in the optimization direction changing for active clients. However, the norm of orthogonal gradient can restrict the deviation of the local model's optimization path. If the difference between the gradient and the orthogonal base becomes large, the direction of the orthogonalized gradient diverges from that of the original gradient, and the norm of the orthogonalized gradient becomes small. The small norm of orthogonal gradient can restrict the updating of the local model, which prevents this local model updating away from its optimal status. If the difference between the gradient and the orthogonal base becomes small, which means that the subtracted projection component becomes small and the updating direction of the gradient after orthogonalization also becomes small, the local model can still update towards its original updating direction. Therefore, **Fed-OGD** does not have impact on the optimization paths of active clients.

### *D. Scalability and Complexity Analysis*

**Caching overhead:** The **Fed-OGD** requires the server to cache the gradients from $K$ clients and the two orthogonal bases. Assuming the size of each client's gradient is $G$ bytes, the total storage requirement on the server is $(K+2) \times G$ bytes. FedAvg caches similar parameters of $K$ local models for averaging aggregation, so its storage requirement is $K \times G$ bytes. Assuming that $K$=50 and the model is Resnet-18, FedAvg requires 2274.40 MB caching space, while **Fed-OGD** requires 2364.31 MB. The caching spaces of both are similar to each other. The caching requirements of other FL models (e.g. SAFA or MIFA) are the same as those of FedAvg. Therefore, the storage overhead does not limit the practicality of **Fed-OGD** for large-scale FL systems.

**Computation overhead:** Compared to FedAvg, the additional computation overhead of **Fed-OGD** is primarily from the orthogonalization. Other computations are the same as FedAvg. For the orthogonalization, each client projects its current gradient onto the orthogonal bases as in Eq. (13), which only calculates the inner product once and the norm once. Hence, the computation overhead of the gradient orthogonalization for each client is $O(|w|)$, where $|w|$ indicates the number of model parameters, which is also equivalent to the number of gradient parameters. While the computation overhead of a training iteration is $O(n \times |x| \times |w|)$ in FedAvg, where $n$ indicates the batch size, $|x|$ denotes the size of a single sample. Because $n \times |x| \gg 1$, the computation overhead of the local training of **Fed-OGD** is $O((n \times |x| + 1) \times |w|) \approx O((n \times |x| \times |w|)$, which is the computation overhead of FedAvg. Therefore, the computation overhead from our **Fed-OGD** is similar to FedAvg, and the computation overhead does not limit the practicality of **Fed-OGD** for large-scale FL systems.

The orthogonalization achieved by **Fed-OGD** holds two steps: First, we calculate the projected component of gradient on orthogonal base, with $proj(g,b) = \frac{\langle g,b \rangle}{\|b\|^2} \cdot b$ where $g$ denotes the gradient and $b$ indicates the orthogonal base. $\langle g,b \rangle$ denote the inner product of gradient, $g$, and orthogonal base $b$, and, $\|b\|$, is the norm of $b$. Last, we subtract the projected component from gradient, $g$, with $g = g - proj(g,b)$. Since the computational complexity of inner product and norm is $O(|g|)$, our computation complexity of **Fed-OGD** is also $O(|g|)$, which is better than Householder transformation and Givens transformation. Moreover, the orthogonal value of **Fed-OGD** is the real value (i.e., $g \cdot b = 0$) rather than the approximation, which is higher than $O(|g|)$ of **Fed-OGD**.

The orthogonalization achieved by the Householder transformation holds three steps: First, we calculate the Householder vector, $v$, with $v = g - (b \cdot g) \cdot b$ where $g$ denotes the gradient and $b$ indicates the orthogonal base. Then, we construct the Householder transformation matrix, $H$, with $H = I - 2\frac{vv^T}{v \cdot v}$ where $I$ indicates a $(|g| \times |g|)$ identity matrix (Assuming the number of the gradient parameters is $|g|$). $v^T$ indicates the transpose of vector, $v$, and $vv^T$ means the outer product of the vector $v$ with its transpose. Last, we apply the Householder transformation to the gradient, $v$, to obtain the orthogonal gradient, $\tilde{g}$, with $\tilde{g} = Hg = g - 2\frac{v(v^T g)}{v^T v}$. Obviously, the computational complexity of Householder transformation is $O(|g|^2)$, due to the outer product.

The orthogonalization achieved by the Givens transformation gradually rotates the gradient until it is orthogonal to the orthogonal base. It includes four steps: First, we calculate the cosine with $c(i,j) = \frac{g_i}{\sqrt{{g_i}^2 + {g_j}^2}}$ and calculate the sine with $s(i,j) = \frac{g_j}{\sqrt{{g_i}^2 + {g_j}^2}}$ where $g$ denotes the gradient and $g_i$ means the $i$-th parameter of $g$. Then, we construct a $(|g| \times |g|)$ identity matrix, $G(i,j)$, where $n$ indicates the number of the model parameters. We replace $(i,i)$-th, $(i,j)$-th, $(j,i)$-th, $(j,j)$-th parameters of $G$ with $c(i,j)$, $s(i,j)$, $-s(i,j)$, $c(i,j)$. Next, we rotate the parameters, $g_i$ and $g_j$, with $[g_i, g_j]^T = G[g_i, g_j]^T$ where $[g_i, g_j]^T$ indicate the transpose of $[g_i, g_j]$. Last, we repeat this process until $g \cdot b \approx 0$. Since the rotation needs $O(|g|^2)$ and each rotation involves $O(|g|)$, the total computational complexity is $O(|g|^3)$. Moreover, since $g \cdot b \approx 0$ is the approximate orthogonal value, the performance of the Givens transformation is further worse than **Fed-OGD**. Therefore, **Fed-OGD** holds more efficient than both Householder and Givens transformations.

### *E. Convergence Analysis*

**Assumption 1.** $\forall k \in [0, \ldots, K]$, the local functions, $f_k$, are all $L$-smooth with $L$>0: $f_k(w) - f_k(v) \leq \langle \nabla f_k(v), w - v \rangle + \frac{L}{2}\|w - v\|^2, \forall w, v \in \mathbb{R}^d$.

**Assumption 2.** The global objective function, $f(\cdot)$, satisfies the condition of $\frac{1}{2}\|\nabla f(w)\|^2 \geq \mu\left(f(w) - f(w^*)\right) \forall w, w^* \in \mathbb{R}^d$.

**Assumption 3.** Considering non-IID data among different clients, a heterogeneity parameter, $\sigma$, satisfies the condition of $\frac{1}{K}\sum_{k=1}^{K}\|\nabla f_k(w_k^{t,\tau}) - \nabla f(w_k^{t,\tau})\|^2 \leq \sigma^2$.

As described in the previous subsection, the gradients uploaded from active clients is orthogonal to the orthogonal basis $b_2$, which is the averaging gradients of stragglers cached in server. Thus we have $\langle \Delta w_i^t, \frac{1}{|M_2|}\sum_{j=1}^{|M_2|}\Delta w_j^{t-s(t,i)} \rangle = 0, i \in [1, |M_1|]$ in which $\frac{1}{|M_2|}\sum_{j=1}^{|M_2|}\Delta w_j^{t-s(t,j)}$ is the averaging gradients from stragglers cached in server, and $\Delta w_j^t$ is the gradient uploaded from $j$-th active client at epoch $t$. With this, we properly state that our **Fed-OGD** can get converged in the following.

**Proposition 2.** If **Assumption 1-3** hold, and the gradients cached in server at $t$-th epoch satisfies $\langle \Delta w_i^t, \frac{1}{|M_2|}\sum_{j=1}^{|M_2|}\Delta w_j^{t-s(t,i)} \rangle = 0, i \in [1, |M_1|]$, **Fed-OGD** gets converged with the number of epoch increasing.

**Proof.** The updating procedure of **Fed-OGD** can be summarized as $w^{t+1} = w^t - \frac{\eta_g}{K}\sum_{k=1}^{K}\Delta w_k^t$ where $\Delta w_k^t = \sum_{\tau=1}^{l}\nabla f_k(w_k^t, \tau)$ and $l$ denote the total number of iterations. If $\|w^0 - w^*\|^2$ is bounded, it indicates that **Fed-OGD** can render the global model, $w^0$, converged to its optimal status, $w^*$. In this way, $\|w^{t+1} - w^*\|^2$ is deduced as:

$$\|w^{t+1} - w^*\|^2 = \left\| w^t - w^* - \frac{\eta_g}{K}\sum_{k=1}^{K}\Delta w_k^t \right\|^2$$

$$= \left\|w^t - w^*\right\|^2 + \underbrace{\frac{\eta_g^2}{K^2}\left\|\sum_{k=1}^{K}\Delta w_k^t\right\|^2}_{\mathcal{T}_1} \underbrace{-2\frac{\eta_g}{K}\sum_{k=1}^{K}\langle \Delta w_k^t, w^t - w^*\rangle}_{\mathcal{T}_2}$$

For the first term $\mathcal{T}_1$, we define $\Delta w_k^t$ as:

$$\frac{\eta_g}{K}\sum_{k=1}^{K}\Delta w_k^t = \frac{\eta_g}{|M_1|}\sum_{i=1}^{|M_1|}\Delta w_i^t + \frac{\eta_g}{|M_2|}\sum_{j=1}^{|M_2|}\Delta w_j^{t-s(t,j)} \tag{5}$$

Based on the square expansion formula, $\mathcal{T}_1$ is deduced as:

$$\begin{aligned}\mathcal{T}_1 &= \left\|\frac{\eta_g}{|M_1|}\sum_{i=1}^{|M_1|}\Delta w_i^t + \frac{\eta_g}{|M_2|}\sum_{j=1}^{|M_2|}\Delta w_j^{t-s(t,j)}\right\|^2 \\ &= \underbrace{\left\|\frac{\eta_g}{|M_1|}\sum_{i=1}^{|M_1|}\Delta w_i^t\right\|^2}_{\mathcal{S}_1} + \underbrace{\left\|\frac{\eta_g}{|M_2|}\sum_{j=1}^{|M_2|}\Delta w_j^{t-s(t,j)}\right\|^2}_{\mathcal{S}_2} \\ &\quad + \underbrace{\frac{2\eta_g^2}{|M_1||M_2|}\langle\sum_{i=1}^{|M_1|}\Delta w_i^t, \sum_{j=1}^{|M_2|}\Delta w_j^{t-s(t,j)}\rangle}_{\mathcal{S}_3}\end{aligned}$$

To prove that the first two terms $\mathcal{S}_1$, $\mathcal{S}_2$ are bounded, it is necessary to prove $\|\nabla f(x)\|^2$ is bounded. Hence, based on the definition of $L$-smoothness and $Jensen$ inequality, we have:

$$\begin{aligned}f^* &\leq f(x - \frac{1}{L}\nabla f(x)) \\ &\leq f(x) + \frac{1}{2L}\|\nabla f(x)\|^2 - \langle \nabla f(x), \frac{1}{L}\nabla f(x)\rangle \\ &= f(x) - \frac{1}{2L}\|\nabla f(x)\|^2\end{aligned}$$

Rearranging the terms on both sides gives $\|\nabla f(x)\|^2 \leq 2L(f(x) - f^*)$. Therefore, the first and the second items of this equation can be deduced as:

$$\begin{aligned}\mathcal{S}_1 &\leq \frac{2L\eta_g^2}{|M_1|^2}\sum_{i=1}^{|M_1|}\sum_{\tau=1}^{l}\left(f_i(w_i^t) - f_i^*\right) \\ \mathcal{S}_2 &\leq \frac{2L\eta_g^2}{|M_2|^2}\sum_{j=1}^{|M_2|}\sum_{\tau=1}^{l}\left(f_j(w_j^{t-s(t,j)}) - f_j^*\right)\end{aligned}$$

It means that $\mathcal{S}_1$ and $\mathcal{S}_2$ are bounded. Based on $\langle\frac{1}{|M_1|}\sum_{i=1}^{|M_1|}\Delta w^t, \Delta w_i^t\rangle = 0, i \in [1, |M_2|]$, which indicates that the uploaded gradients of active clients at $t$-th epoch is orthogonal to the orthogonal basis of stragglers, we have:

$$\mathcal{S}_3 = \frac{2\eta_g^2}{|M_1|}\sum_{j=1}^{|M_2|}\langle \Delta w_i^t, \frac{1}{|M_2|}\sum_{j=1}^{|M_2|}\Delta w_j^{t-s(t,i)}\rangle = 0$$

thus we find that $\mathcal{S}_3 = 0$ and $\mathcal{S}_3$ is bounded. With this, $\mathcal{T}_1$ is bounded as well.

For the second term $\mathcal{T}_2$, we transform it with inequalities from the **Assumption 2**:

$$\begin{aligned}\mathcal{T}_2 &= \frac{2\eta_g}{K}\sum_{k=1}^{K}\sum_{\tau=1}^{l}\langle \nabla f(w_k^{t,\tau}), w^t - w^*\rangle \\ &\quad + \frac{2\eta_g}{K}\sum_{k=1}^{K}\sum_{\tau=1}^{l}\langle \nabla f_k(w_k^{t,\tau}) - \nabla f(w_k^{t,\tau}), w^t - w^*\rangle \\ &\geq \frac{2\eta_g}{K}\sum_{k=1}^{K}\sum_{\tau=1}^{l} f(w_k^{t,\tau}) - f(w^*) + \frac{\mu}{2}\|w_k^{t,\tau} - w^*\|^2 \\ &\quad + \frac{2\eta_g}{K}\sum_{k=1}^{K}\sum_{\tau=1}^{l}\langle \nabla f_k(w_k^{t,\tau}) - \nabla f(w_k^{t,\tau}), w^t - w^*\rangle\end{aligned}$$

Because $w_k^{t,\tau}$ is local model parameters, let $\|w_k^{t,\tau} - w^t\| \leq \delta$. Then, we expand $\mathcal{T}_2$ by Cauchy-Schwarz inequality and **Assumption 3** to:

$$\begin{aligned}\mathcal{T}_2 &\geq \frac{2\eta_g}{K}\sum_{k=1}^{K}\sum_{\tau=1}^{l} f(w_k^{t,\tau}) - f(w^*) + \frac{\mu}{2}\|w_k^{t,\tau} - w^*\|^2 \\ &\quad + \frac{2\eta_g}{K}\sum_{k=1}^{K}\sum_{\tau=1}^{l}\|\nabla f_k(w_k^{t,\tau}) - \nabla f(w_k^{t,\tau})\| \cdot \|w^t - w^*\| \\ &\geq \frac{2\eta_g}{K}\sum_{k=1}^{K}\sum_{\tau=1}^{l} f(w_k^{t,\tau}) - f(w^*) \\ &\quad + \frac{\mu}{2}\|w_k^{t,\tau} - w^*\|^2 - 2\eta_g l\sigma\|w^t - w^*\| \\ &\geq 2\eta_g l\left(f(w^t) - f(w^*)\right) - 2\eta_g l(L\delta + \sigma)\|w^t - w^*\|\end{aligned}$$

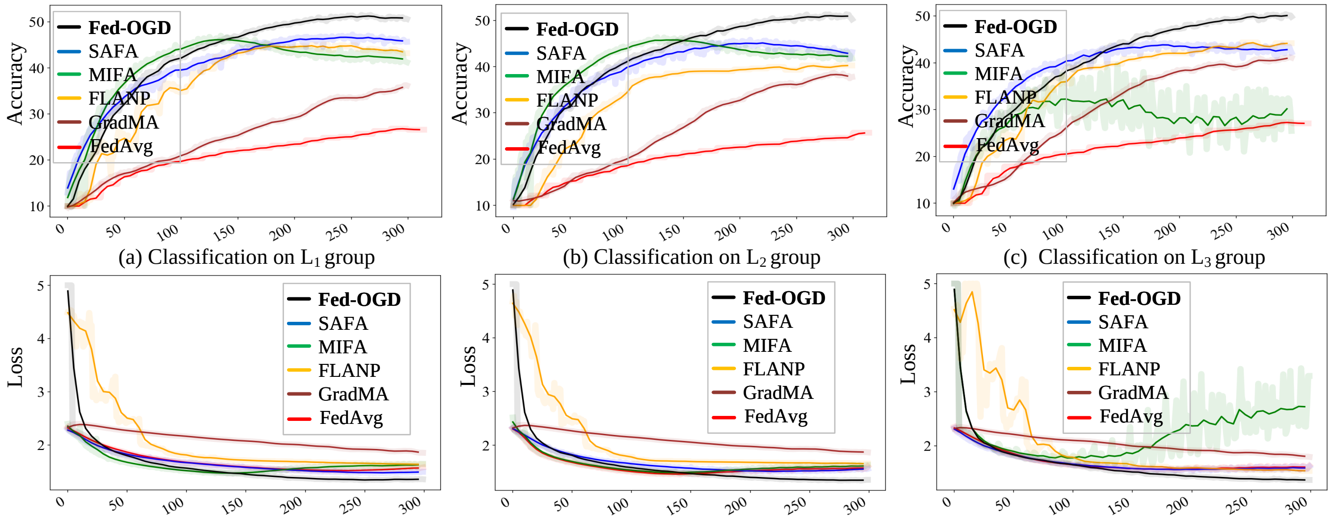
Here, we get $\mathcal{T}_2$ bounded. Under such a scenario, which $\mathcal{T}_1$ and $\mathcal{T}_2$ is bounded, we get $\|w^{t+1} - w^*\|^2 - \|w^t - w^*\|^2$ is also bounded. In other words, our **Fed-OGD** gets converged after $t$ epochs if both **Assumption 1-3** hold.

# IV. Experiment

## A. Experiment Setting

For the experiments, four public datasets are studied, which are CIFAR-10, CIFAR-100, Tiny-ImageNet, and AG_NEWS. Because the straggler issue rarely appears in IID (Independent and identically distributed) scenario and it is often in non-IID scenario [37], [38], all models are validated on non-IID scenarios (i.e., few categories hold many samples while many categories have few samples in a client).

Five SOTA FL models are employed, which are **FedAvg** [1], **SAFA** [22], **MIFA** [21], **FLANP** [39], and **GradMA** [31]. These models are representatives in reducing straggler issue. We validate those models on the four datasets, and set three different non-IID scenarios for each dataset. Moreover, we employ 50 clients and 1 server for all cases. After that, we divide all clients into three groups (one active client group and two straggler groups), according to their responses to server at each epoch. Here, we assume that the responded time of active client group is $r$, while that of two straggler groups are $3r$ and $5r$, in which $r$ indicates responding server at each epoch and $3r$ ($5r$) denotes responding to server every 3 (5) epochs. Besides, we set the proportions of each



(a) Classification on $L_1$ group (b) Classification on $L_2$ group (c) Classification on $L_3$ group

(d) Convergence on $L_1$ group (e) Convergence on $L_2$ group (f) Convergence on $L_3$ group

Fig. 2: The classification performance of all models on CIFAR-10 dataset with non-IID scenario. Sub-figures (a)-(c) show the classification accuracy, while sub-figures (d)-(f) show the convergence of all models.

group as follows: $L_1$={$r$:60%, 3$r$:20%, 5$r$:20%}, $L_2$={$r$:40%, 3$r$:30%, 5$r$:30%}, $L_3$={$r$:20%, 3$r$:40%, 5$r$:40%}. Take $L_1$ as an example, it indicates that the active client group takes 60% proportion of all clients, while each straggler group takes 20% proportion of all clients, respectively.

TABLE I: Accuracy variances on CIFAR-10 at different cases

| CIFAR-10 | $L_1$ | $L_2$ | $L_3$ |
|---|---|---|---|
| **Fed-OGD** | **9.552**$\times$ $10^{-5}$ | **1.942**$\times$ **10**$^{-5}$ | **9.233**$\times$**10**$^{-4}$ |
| FedAvg | $1.726\times10^{-3}$ | $1.046\ \times10^{-3}$ | $2.700\times10^{-3}$ |
| SAFA | $1.614\times10^{-4}$ | $3.666\ \times10^{-3}$ | $1.849\times10^{-3}$ |
| MIFA | $1.643\times10^{-3}$ | $2.997\ \times10^{-4}$ | $2.440\times10^{-3}$ |
| FLANP | $2.535\times10^{-4}$ | $1.326\ \times10^{-3}$ | $2.562\times10^{-3}$ |
| GradMA | $1.524\times10^{-2}$ | $3.161\ \times10^{-3}$ | $3.081\times10^{-3}$ |

To quantitatively evaluate the performance of all models, the **Top-1 accuracy** [1] is employed as the evaluation metric where a higher score indicates the better global model performance. Besides, the stragglers heavily affect the robust of the global model, it is necessary to compare the robustness of **Fed-OGD** with that of baselines. In this way, **Accuracy variance** is employed [40], [41], with $var = \frac{1}{2}\left(|a_1 - a|^2 + |a_2 - a|^2\right)$ where $a$ indicates the averaging accuracy from the global model and $a_1$ denotes the accuracy from active clients as well as $a_2$ represents the accuracy from stragglers. The smaller the variance is, the better robustness the global model holds.

### *B. CIFAR-10*

For the CIFAR-10, 50000 training samples are divided into 50 sub-sets, and each sub-set has 1000 images with all categories. In each sub-set, there is only one main category, which takes the 95% proportion of samples, and the remaining images are from other categories. For example, sub-set 1 contains 950 car images (i.e., $\frac{950}{1000}$), and sub-set 2 includes 950 horse images. The rest of each sub-set holds other 9 categories. One client solely holds one sub-set, and we employ a Resnet-18 model to each client.

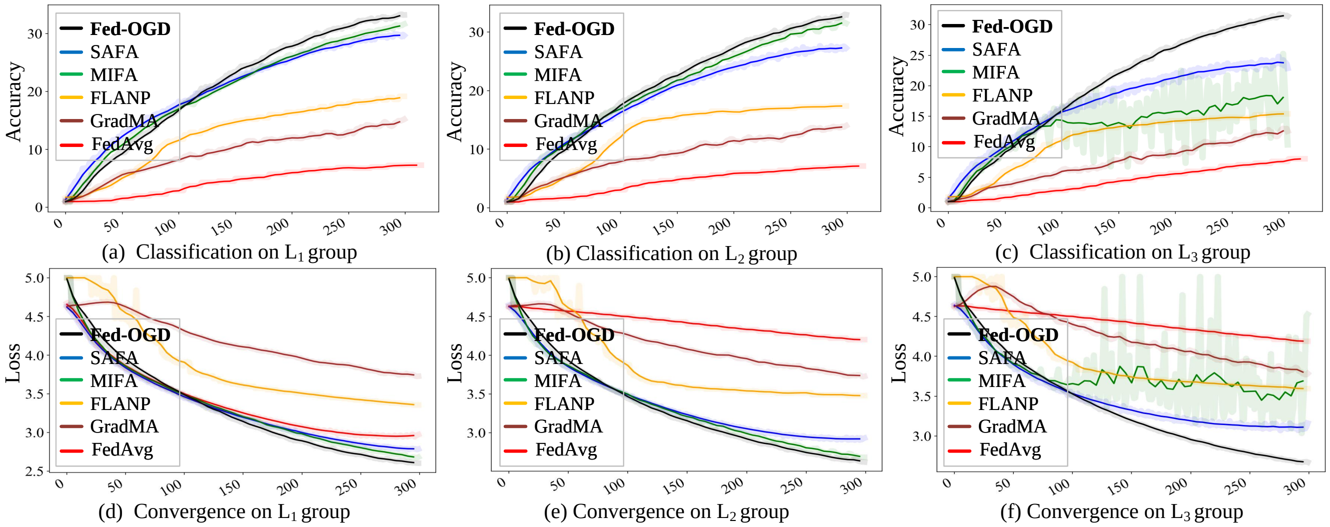
The experimental results are shown in Fig. 2 in which sub-figures (a)-(c) show the classification accuracy results in different proportions of groups. Obviously, our **Fed-OGD** outperforms other SOTA FL models in all cases, especially in $L_2$ case that **Fed-OGD** improves accuracy by 15.79% compared to the SAFA model that holds a highest accuracy score among all baseline models. Moreover, only the accuracy score from **Fed-OGD** is over 50% (i.e., 51.49%), while others are less than 47%. Sub-figures (d)-(f) shows the convergences of all models. From the three sub-figures, we can observe that **Fed-OGD** gets better converged among all models. The recent models, such as FLANP and GradMA, are difficult to get converged, in which FLANP holds fluctuation while GradMA cannot get converged in the three cases. MIFA model cannot get converged in the $L_3$ case, and it becomes worse when epoch increases. Those results demonstrate the effectiveness of our **Fed-OGD** in reducing the straggler issue.

To describe the robustness of **Fed-OGD** that is better than baseline models, we compare its variance with SOTA baseline models as listed in Table I. From Table I, it is obviously observed that **Fed-OGD** holds the smallest variance value over all baseline models. The smallest variance indicates that **Fed-OGD** holds the best robust performance in all cases, which further demonstrates the effectiveness of our **Fed-OGD**.

### *C. CIFAR-100*

CIFAR-100 also holds 50000 training images and 10000 test images. Since there are 100 categories, each sub-set holds 2 main categories. In this way, we set the proportions of each main category holds 40% proportion of all samples in a sub-set. Similar to CIFAR-10, each client solely holds one sub-set and one Resnet-18 model.

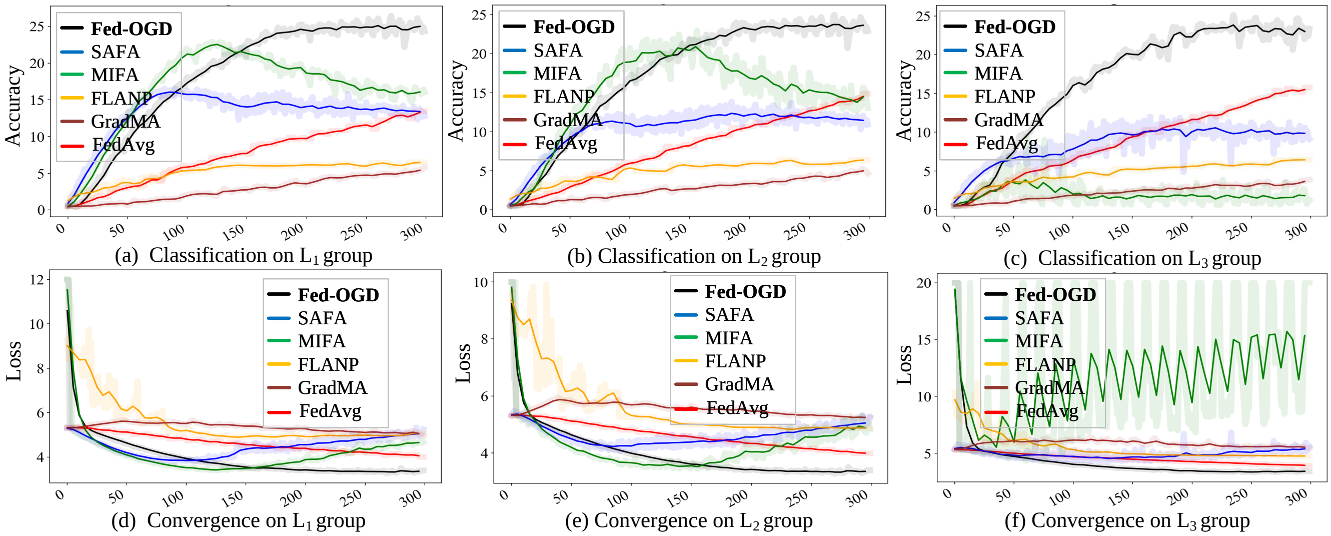
The experimental results are shown in Fig. 3. From Fig. 3, we can observe that **Fed-OGD** still obtains the best classification performance over all baseline models in all cases. Sub-figures (a)-(c) show that **Fed-OGD** achieves the highest accuracy score in all models. The FedAvg holds the lowest



(a) Classification on $L_1$ group (b) Classification on $L_2$ group (c) Classification on $L_3$ group

(d) Convergence on $L_1$ group (e) Convergence on $L_2$ group (f) Convergence on $L_3$ group

Fig. 3: The classification performance of all models on CIFAR-100 dataset with non-IID scenario. Sub-figures (a)-(c) show the classification accuracy, while sub-figures (d)-(f) show the convergence of all models.



(a) Classification on $L_1$ group (b) Classification on $L_2$ group (c) Classification on $L_3$ group

(d) Convergence on $L_1$ group (e) Convergence on $L_2$ group (f) Convergence on $L_3$ group

Fig. 4: The classification performance of all models on Tiny-ImageNet dataset. Sub-figures (a)-(c) show the classification accuracy, while sub-figures (d)-(f) show the convergence of all models.

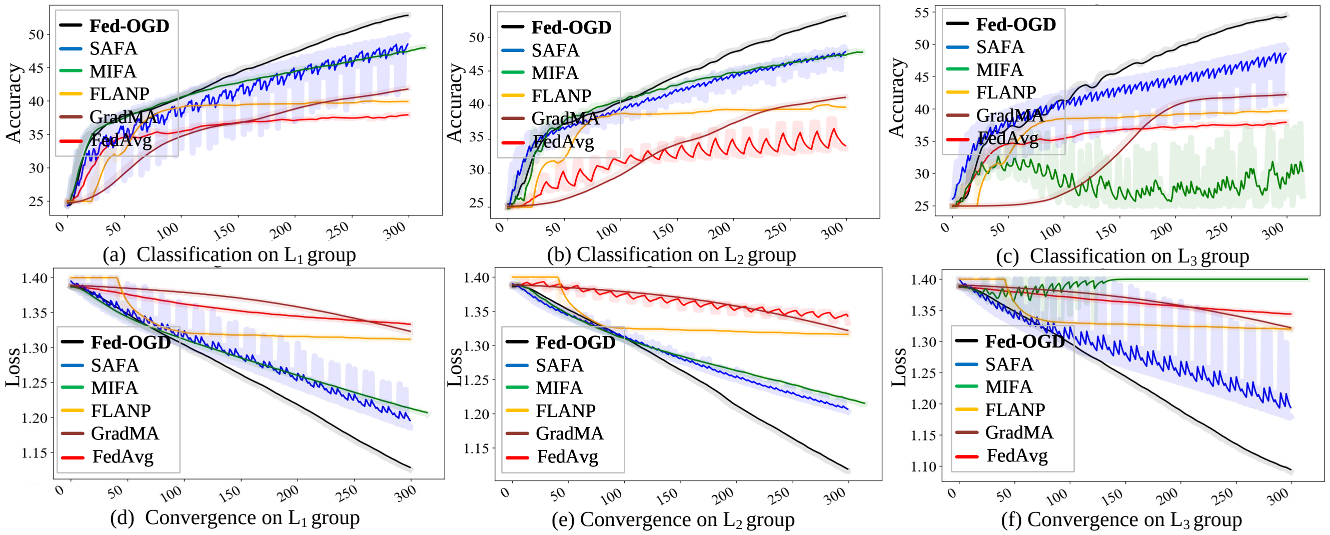
TABLE II: Accuracy variances on CIFAR-100 at different cases

| CIFAR-100 | $L_1$ | $L_2$ | $L_3$ |
|---|---|---|---|
| **Fed-OGD** | $\mathbf{1.408\times 10^{-5}}$ | $\mathbf{3.045\times 10^{-5}}$ | $\mathbf{1.150\times 10^{-5}}$ |
| FedAvg | $2.004\times 10^{-4}$ | $5.895\times 10^{-5}$ | $3.239\times 10^{-4}$ |
| SAFA | $2.630\times 10^{-4}$ | $1.563\times 10^{-3}$ | $1.923\times 10^{-3}$ |
| MIFA | $9.990\times 10^{-5}$ | $1.619\times 10^{-4}$ | $4.589\times 10^{-5}$ |
| FLANP | $4.074\times 10^{-5}$ | $1.882\times 10^{-4}$ | $8.254\times 10^{-4}$ |
| GradMA | $1.958\times 10^{-2}$ | $4.720\times 10^{-3}$ | $1.226\times 10^{-2}$ |

accuracy score, while the recent studies (e.g. FLANP and GradMA) achieve the lower accuracy score in the three cases. Although the accuracy score from MIFA approaches to that from **Fed-OGD** in both $L_1$ and $L_2$ cases (MIFA gets 31.65% and **Fed-OGD** achieves 33.35% in $L_1$ case while MIFA gets 31.5% and **Fed-OGD** achieves 33.0% in $L_2$ case), it decreases sharply in $L_3$ case (MIFA obtains 25.27% and **Fed-OGD** reaches to 31.59%). Sub-figures (d)-(f) show the convergence performance of all models. From the three sub-figures, it is observed that **Fed-OGD** gets better convergence than other baseline models, especially in $L_3$ case in which MIFA cannot get converged and it is always fluctuation. The loss from FedAvg decreases very slowly, which means that FedAvg cannot get converged. FLANP, GradMA, and SAFA hold larger loss values than **Fed-OGD**, which indicates a worse convergence. Moreover, we compare the robustness of **Fed-OGD** with baseline models as shown in Table II. It is observed that **Fed-OGD** still holds the best robust performance over baseline models in all cases, which further demonstrates the effectiveness of **Fed-OGD** in reducing the straggler issue.

### *D. Tiny-ImageNet*

We continue to validate all models on Tiny-ImageNet with Resnet-34, and the experimental results are shown in Fig. 4. From Fig. 4 (a)-(c), we still observe that our **Fed-OGD**



(a) Classification on $L_1$ group (b) Classification on $L_2$ group (c) Classification on $L_3$ group

(d) Convergence on $L_1$ group (e) Convergence on $L_2$ group (f) Convergence on $L_3$ group

Fig. 5: The classification performance of all models on AG_NEWS dataset. Sub-figures (a)-(c) show the classification accuracy, while sub-figures (d)-(f) show the convergence of all models.

TABLE III: Accuracy variances on Tiny-ImageNet at different cases

| Tiny-ImageNet | $L_1$ | $L_2$ | $L_3$ |
|---|---|---|---|
| **Fed-OGD** | $\mathbf{5.349\times 10^{-6}}$ | $\mathbf{1.594\times 10^{-5}}$ | $\mathbf{5.221\times 10^{-8}}$ |
| FedAvg | $1.596\times10^{-4}$ | $6.269\times10^{-4}$ | $1.471\times10^{-4}$ |
| SAFA | $2.721\times10^{-3}$ | $1.105\times10^{-3}$ | $1.375\times10^{-2}$ |
| MIFA | $2.184\times10^{-5}$ | $1.610\times10^{-4}$ | $1.650\times10^{-5}$ |
| FLANP | $8.715\times10^{-4}$ | $1.404\times10^{-3}$ | $3.763\times10^{-3}$ |
| GradMA | $3.219\times10^{-4}$ | $1.748\times10^{-3}$ | $4.133\times10^{-4}$ |

TABLE IV: Accuracy variances on AG_NEWS at different cases

| AG_NEWS | $L_1$ | $L_2$ | $L_3$ |
|---|---|---|---|
| **Fed-OGD** | $\mathbf{5.349\times 10^{-6}}$ | $\mathbf{1.594\times 10^{-5}}$ | $\mathbf{5.221\times 10^{-8}}$ |
| FedAvg | $1.596\times10^{-4}$ | $6.269\times10^{-4}$ | $1.471\times10^{-4}$ |
| SAFA | $2.721\times10^{-3}$ | $1.105\times10^{-3}$ | $1.375\times10^{-2}$ |
| MIFA | $2.184\times10^{-5}$ | $1.610\times10^{-4}$ | $1.650\times10^{-5}$ |
| FLANP | $8.715\times10^{-4}$ | $1.404\times10^{-3}$ | $3.763\times10^{-3}$ |
| GradMA | $3.219\times10^{-4}$ | $1.748\times10^{-3}$ | $4.133\times10^{-4}$ |

achieves the largest classification accuracy in $L_1$-$L_3$ cases. The recent studies, such as FLANP and GradMA, obtain accuracy scores with around of 5% in the three cases. MIFA is unstable (See sub-figures (a) and (b)), and its output score decreases after 150 epochs. As to FedAvg and SAFA, their scores are less than **Fed-OGD**. Sub-figures (d)-(f) show the convergence performance of all models. It is easy observed that **Fed-OGD** gets better convergence than all baseline models. The loss values from both FedAvg and GradMA decrease slowly, which means that they are difficult to get converged. MIFA gets intense fluctuation (See sub-figure (f)), which indicates that MIFA cannot get convergence. Besides, we also compare the robustness of **Fed-OGD** with all baseline models as the previous two datasets. The compared results are shown in Table III. Obviously, our **Fed-OGD** still achieves the smallest variance values, holding the best robust performance. This also demonstrates the effectiveness of **Fed-OGD** in reducing the straggler issue.

### *E. AG_NEWS*

To validate the effectiveness of our **Fed-OGD** in NLP task, we compare it with baseline models on the text dataset AG_NEWS. To make a fair comparison, we employ the TextCNN model for all FL models and other experimental settings are unchanged as other datasets. Fig. 5 (a)-(c) show that **Fed-OGD** still achieves the highest accuracy scores across $L_1$-$L_3$ cases. It is over 50% accuracy score, while the accuracy scores of other FL models are lower than 50%. Moreover, the accuracy of FedAvg is the lowest in both $L_1$ and $L_2$ cases, and MIFA performs worst in $L_3$ case, and their scores are below 40%. Note that the recent studies, such as FLANP and SAFA, also lag far behind **Fed-OGD**. Fig. 5 (d)-(f) show the convergence behavior of all models. **Fed-OGD** demonstrates the best convergence over all baselines. FedAvg and GradMA show the worst convergence in $L_1$-$L_3$ cases, and MIFA becomes instable after 50 epochs (see sub-figure (f)) in $L_3$ case. As the other datasets, we also evaluate the robustness of **Fed-OGD** on AG_NEWS dataset, and the evaluated results are shown in Table IV. Obviously, the variance values of **Fed-OGD** are still lower than those of other baseline models, further demonstrating the effectiveness of **Fed-OGD** in addressing the straggler issue in NLP task.

The communication overhead in FL is from the mutual communication between the server and all clients, e.g. the clients uploading the gradients and the server broadcasting the parameters of the global model to local models. For the AG_NEWS dataset, we employ 50 clients, and each client holds a TextCNN model (9.92 MB). The experimental results are shown in Fig. 5. From sub-figure (d) of Fig. 5, it is observed that **Fed-OGD** achieves the fastest convergence, reaching a loss value of 1.219 at 193 epoch. However, SAFA, which has the second-fastest convergence, reaches the same

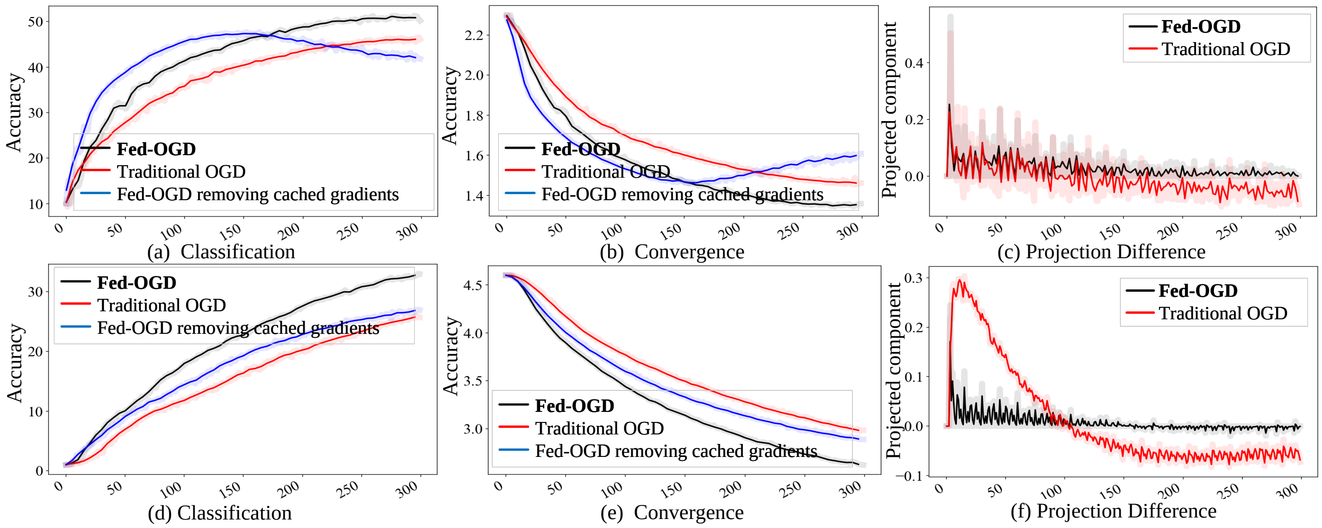


Fig. 6: The ablation studies on both CIFAR-10 and CIFAR-100 datasets in $L_2$ case. Sub-figures (a)-(c) show the classification performance on CIFAR-10 dataset, while sub-figures (d)-(f) show the classification performance on CIFAR-100 dataset.

loss value at 299 epoch. For **Fed-OGD**, this communication overhead is $9.92 \times 3 = 29.76$ MB, given that **Fed-OGD** holds the extra orthogonal bases. With this, the total communication overhead at 193 epoch is $29.76 \times 193 \times (\frac{30}{1} + \frac{10}{3} + \frac{10}{5}) = 202943.36$ MB in $L_1$ case, while the total communication overhead at 299 epoch for SAFA is $19.84 \times 299 \times (\frac{30}{1} + \frac{10}{3} + \frac{10}{5})$=209602.986 MB. Other baseline models have the higher communication overhead than SAFA. Therefore, our **Fed-OGD** achieves smaller communication overhead than the baseline models. The similar scenarios are also shown in other datasets. Take CIFAR10 as an example. **Fed-OGD** reaches the loss value of 1.497 at 124 epoch, while SAFA, which gets the second-fastest convergence among the baseline models, reaches the same loss at 299 epoch. **Fed-OGD** holds $44.70 \times 124 \times (\frac{30}{1} + \frac{10}{3} + \frac{10}{5}) = 195845.59$ MB, while SAFA obtains $44.70 \times 299 \times (\frac{30}{1} + \frac{10}{3} + \frac{10}{5}) = 472240.60$ MB, given that ResNet-18 has size of 44.70 MB. Therefore, Obviously, the total communication overhead of our **Fed-OGD** is smaller than the baseline models.

### *F. Ablation Studies*

Different from the traditional OGD, **Fed-OGD** caches the latest gradients in server for global model aggregation at each epoch and views the latest cached gradients of stragglers as orthogonal bases for active clients, and also views the last gradients of active clients as orthogonal bases for stragglers. Therefore, we conduct a series of ablation studies to demonstrate the necessity of each component of **Fed-OGD** by only removing cached gradients and directly replacing our OGD with the traditional OGD. Here, we take $L_2$ case and both CIFAR-10 and CIFAR-100 datasets as an example. The experimental results are shown in Fig. 6 (a) and (d). It illustrates that each component is necessary to **Fed-OGD**, because removing any component could reduce performance. Since **Fed-OGD** relies on the projected component to perform an orthogonal operation to guide the global model towards its optimal status, we illustrate the projected components of both **Fed-OGD** and the traditional OGD as shown in the sub-figures (c) and (f) of Fig. 6. The larger the projected component, the smaller the difference between the orthogonal bases of the straggler clients and the current gradients of the active clients, and the better convergence the global model gets (see Fig. 6 (b) and (e)). The difference curves show the importance of our idea, which is better than the traditional OGD method. Moreover, the variance of traditional OGD is $2.853 \times 10^{-5}$ ($1.897 \times 10^{-3}$) and that of **Fed-OGD** removing cached gradients is $6.044 \times 10^{-4}$ ($1.930 \times 10^{-3}$) on CIFAR-10 (CIFAR-100). The variances of **Fed-OGD**, $1.942 \times 10^{-5}$ for CIFAR-10 and $3.045 \times 10^{-5}$ for CIFAR-100, are lower than those variances, further illustrating that each component is important for the robustness of **Fed-OGD**.

### *G. Discussion*

Note that the proportions of each group (i.e., $L_1$, $L_2$, and $L_3$) correspond to three main aspects. In $L_1$ case, the active clients take the largest proportion of all clients, while straggler clients occupy the minority proportion of all clients; in $L_2$ case, the proportion of active clients approaches that of stragglers in all clients; the case of $L_3$ is opposite to $L_1$, active clients occupy the minority proportion of all clients, while straggler clients take most proportion of all clients. Given that the stragglers fail to upload their parameters to server at some epochs, the optimizing direction of the global model is inevitably towards that of the active clients, because the interval uploaded parameters of stragglers have a small impact on the global model, resulting in its convergence difficulty. This is shown in the convergence performance of FedAvg in Fig. 2, Fig. 3, Fig. 4 and Fig. 5 (d)-(f). Moreover, stragglers bring the non-robustness for the global model [8], so we utilize the **variance** metric to quantitatively compare the robustness of **Fed-OGD** with baseline models (see Table I-Table IV).

Given that FL algorithms are usually validated by classification [1], we employ it to demonstrate the performance of each model. From the experimental results (see Fig. 2-Fig. 5), we can observe that **Fed-OGD** not only achieves the highest accuracy score but also has the best convergence among all SOTA FL models. Moreover, we conduct a serious of ablation studies by only removing the cached gradients and replacing our OGD with the traditional OGD to demonstrate the necessity of each component of **Fed-OGD**, which is shown in Fig. 6 (a)-(b) and Fig. 6 (d)-(e). It illustrates the importance of each component, because reducing any one component could bring about a decrease in performance. Besides, we use projected component to measure the difference in optimizing direction as shown in Fig. 6 (c) and (f), which further demonstrate the effectiveness of our **Fed-OGD**. Since most real-world FL classification applications (e.g. medical image analysis [42] and agricultural image analysis [43]) focus on CNN-based model [44], our paper also employs the CNN-based model (e.g. Resnet-18, Resnet-34 and TextCNN) to validate our idea.

## V. Conclusions

In this paper, to deal with the straggler issue, we propose the **Fed-OGD**, which caches the latest gradients of stragglers in server and orthogonalizes the difference between the orthogonal bases (i.e., the latest cached gradients) of stragglers and the current gradients of active clients. The contributions of our **Fed-OGD** are: 1) innovatively explaining the global model convergence difficulty with CF theory and prove it theoretically; 2) presenting a new strategy to guide the global model towards its optimal status rather than the optimial status of active clients; 3) presenting 16.66% (5.37%, 38.51%, and 16.30%) higher classification accuracy and achieving 93.52% (74.94%, 99.68%, and 99.69%) lower variance on CIFAR-10 (CIFAR-100, Tiny-ImageNet, and AG_NEWS) when comparing with the model that holds the best performance in all baselines; 4) providing new insights into the understanding of straggler issue in FL.

## Acknowledgment

This work was supported by the National Key Research and Development Program of China (2022YFF1101100), and was supported by the National Natural Science Foundation of China under Grant Nos. 62372324 and 62032017.

## References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and stat.* PMLR, 2017, pp. 1273–1282.

[2] D. Wang, J. Ren, Z. Wang, Y. Wang, and Y. Zhang, "Privaim: A dual-privacy preserving and quality-aware incentive mechanism for federated learning," *IEEE Transactions on Computers*, 2022.

[3] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[4] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, "Privacy preservation in federated learning: An insightful survey from the gdpr perspective," *Computers & Security*, vol. 110, p. 102402, 2021.

[5] S. Chen, Y. Wang, D. Yu, J. Ren, C. Xu, and Y. Zheng, "Privacy-enhanced decentralized federated learning at dynamic edge," *IEEE Transactions on Computers*, 2023.

[6] Z. Shen, Q. Tang, T. Zhou, Y. Zhang, Z. Jia, D. Yu, Z. Zhang, and B. Li, "Ashl: An adaptive multi-stage distributed deep learning training scheme for heterogeneous environments," *IEEE Transactions on Computers*, 2023.

[7] H. Wang, S. Guo, B. Tang, R. Li, Y. Yang, Z. Qu, and Y. Wang, "Heterogeneity-aware gradient coding for tolerating and leveraging stragglers," *IEEE Transactions on Computers*, vol. 71, no. 4, pp. 779–794, 2021.

[8] 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.

[9] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in neural information processing systems*, vol. 30, 2017.

[10] D. Wu, R. Ullah, P. Harvey, P. Kilpatrick, I. Spence, and B. Varghese, "Fedadapt: Adaptive offloading for iot devices in federated learning," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 20 889–20 901, 2022.

[11] Y.-J. Liu, G. Feng, H. Du, Z. Qin, Y. Sun, J. Kang, and D. Niyato, "Straggler-aware federated learning based on adaptive clustering to support edge intelligence," in *ICC 2024-IEEE International Conference on Communications*. IEEE, 2024, pp. 1867–1872.

[12] Y. Chen, X.-H. Sun, and M. Wu, "Algorithm-system scalability of heterogeneous computing," *Journal of Parallel and Distributed Computing*, vol. 68, no. 11, pp. 1403–1412, 2008.

[13] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.

[14] C. Zhou, H. Tian, H. Zhang, J. Zhang, and J. Jia, "Tea-fed: time-efficient asynchronous federated learning for edge computing," in *CF '21: Computing Frontiers Conference*, 2021.

[15] Y. Zou, S. Shen, M. Xiao, P. Li, D. Yu, and X. Cheng, "Value of information: A comprehensive metric for client selection in federated edge learning," *IEEE Transactions on Computers*, 2024.

[16] D. Yao, W. Pan, Y. Dai, Y. Wan, X. Ding, C. Yu, H. Jin, Z. Xu, and L. Sun, "Fedgkd: Towards heterogeneous federated learning via global knowledge distillation," *IEEE Transactions on Computers*, 2023.

[17] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[18] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *2019-2019 IEEE international conference on communications*. IEEE, 2019, pp. 1–7.

[19] Z. Chai, A. Ali, S. Zawad, S. Truex, A. Anwar, N. Baracaldo, Y. Zhou, H. Ludwig, F. Yan, and Y. Cheng, "Tifl: A tier-based federated learning system," in *Proceedings of the 29th international symposium on high-performance parallel and distributed computing*, 2020, pp. 125–136.

[20] G. Li, Y. Hu, M. Zhang, J. Liu, Q. Yin, Y. Peng, and D. Dou, "Fedhisyn: A hierarchical synchronous federated learning framework for resource and data heterogeneity," in *Proceedings of the 51st International Conference on Parallel Processing*, 2022, pp. 1–11.

[21] X. Gu, K. Huang, J. Zhang, and L. Huang, "Fast federated learning in the presence of arbitrary device unavailability," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 052–12 064, 2021.

[22] W. Wu, L. He, W. Lin, R. Mao, C. Maple, and S. Jarvis, "Safa: A semi-asynchronous protocol for fast federated learning with low overhead," *IEEE Transactions on Computers*, vol. 70, no. 5, pp. 655–668, 2020.

[23] R. Ratcliff, "Connectionist models of recognition memory: constraints imposed by learning and forgetting functions." *Psychological review*, vol. 97, no. 2, p. 285, 1990.

[24] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[25] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal gradient descent for continual learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 3762–3773.

[26] A. Mathiasen, F. Hvilshøj, J. R. Jørgensen, A. Nasery, and D. Mottin, "Faster orthogonal parameterization with householder matrices," in *ICML, Workshop Proceedings*, 2020.

[27] V. Volfson, "Converting of algebraic diophantine equations to a diagonal form with the help of an integer non-orthogonal transformation, maintaining the asymptotic behavior of the number of its integer solutions," *arXiv preprint arXiv:1708.01499*, 2017.

[28] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.

[29] 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, pp. 4229–4238, 2019.

[30] 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.

[31] K. Luo, X. Li, Y. Lan, and M. Gao, "Gradma: A gradient-memory-based accelerated federated learning with alleviated catastrophic forgetting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3708–3717.

[32] X. Li, Z. Qu, B. Tang, and Z. Lu, "Stragglers are not disaster: A hybrid federated learning algorithm with delayed gradients," *arXiv preprint arXiv:2102.06329*, 2021.

[33] S. Lei and D. Tao, "A comprehensive survey of dataset distillation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[34] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi, "Rainbow memory: Continual learning with a memory of diverse samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8218–8227.

[35] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and S. Y. Philip, "Generalizing to unseen domains: A survey on domain generalization," *IEEE transactions on knowledge and data engineering*, vol. 35, no. 8, pp. 8052–8072, 2022.

[36] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural computation*, vol. 10, no. 2, pp. 251–276, 1998.

[37] W. Liu, X. Xu, L. Wu, L. Qi, A. Jolfaei, W. Ding, and M. R. Khosravi, "Intrusion detection for maritime transportation systems with batch federated aggregation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2503–2514, 2022.

[38] W. Li, J. Chen, Z. Wang, Z. Shen, C. Ma, and X. Cui, "Ifl-gan: Improved federated learning generative adversarial network with maximum mean discrepancy model aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[39] A. Reisizadeh, 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.

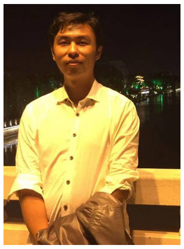
[40] Z. Chai, Y. Chen, A. Anwar, L. Zhao, Y. Cheng, and H. Rangwala, "Fedat: A high-performance and communication-efficient federated learning system with asynchronous tiers," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–16.

[41] C. Yang, M. Xu, Q. Wang, Z. Chen, K. Huang, Y. Ma, K. Bian, G. Huang, Y. Liu, X. Jin *et al.*, "Flash: Heterogeneity-aware federated learning at scale," *IEEE Transactions on Mobile Computing*, 2022.

[42] R. N. Sutton and E. L. Hall, "Texture measures for automatic classification of pulmonary disease," *IEEE Transactions on Computers*, vol. 100, no. 7, pp. 667–676, 1972.

[43] A. Menshchikov, D. Shadrin, V. Prutyanov, D. Lopatkin, S. Sosnin, E. Tsykunov, E. Iakovlev, and A. Somov, "Real-time detection of hogweed: Uav platform empowered by deep learning," *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1175–1188, 2021.

[44] W. Li, Z. Shen, X. Liu, M. Wang, C. Ma, C. Ding, and J. Cao, "Representative kernels-based cnn for faster transmission in federated learning," *IEEE Transactions on Mobile Computing*, 2024.



**Dr. Wei Li** (Member, IEEE) was undergraduate from the subject of Information and Computing Science in 2008, and receives his Master's degree in Agricultural Engineering from South China Agricultural University in 2012, and gets his Ph.D. degree in Software Engineering from Wuhan University in 2019. He is an Associate Professor of School of Artificial Intelligence and Computer Science at Jiangnan University. He was the visiting student of University of Massachusetts Boston and had visited the The Hong Kong Polytechnic University as Research Assistant and Research Fellow. His research interests include Data Mining, Artificial Intelligence, and Federated Learning, and has been published in top-tier journals such as IEEE TMC, TNNLS, TCYB, TCBB, TCSVT, T-ASE, ACM TOMM, etc.

**Mr. Zicheng Shen** received the B.S. degree in Applied Chemistry from Jiangnan University in 2022. Currently, he is working toward the M.S. degree with the School of Artifcial Intelligence and Computer Science, Jiangnan University. His research interests mainly include software engineering and federated learning.

**Dr. Xiulong Liu** is currently a professor in College of Intelligence and Computing, Tianjin University, China. Before that, he received the B.E. and Ph.D. degrees from Dalian University of Technology (China) in 2010 and 2016, respectively. He also worked as a visiting researcher in Aizu University, Japan; a postdoctoral fellow in The Hong Kong Polytechnic University, Hong Kong; and a postdoctoral fellow in the School of Computing Science, Simon Fraser University, Canada. His research interests include wireless sensing and communication, indoor localization, and networking, etc. His research papers were published in many prestigious journals and conferences including TON, TMC, TC, TPDS, TCOM, INFOCOM, and ICNP, etc. He received Best Paper Awards from ICA3PP 2014 and IEEE System Journal 2017. He is also the recipient of CCF Outstanding Doctoral Dissertation award 2017.

**Dr. Chuntao Ding** (Member, IEEE) received his Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2021. He is currently a lecturer at Beijing Jiaotong University. He also worked as a research assistant with Hong Kong Polytechnic University and as a visiting scholar at Michigan State University. His research interests include service computing, edge computing, and multi-task learning. He received the Outstanding Ph.D. Thesis award of IEEE Technical Committee on Cloud Computing. He has published more than 20 top conference and journal papers, such as IEEE CVPR, IEEE TPDS, IEEE TMC, etc.

**Dr. Jiaxing Shen** is an Assistant Professor with the School of Data Science at Lingnan University. He received the B.E. degree in Software Engineering from Jilin University in 2014, and the Ph.D. degree in Computer Science from the Hong Kong Polytechnic University in 2019. He was a visiting scholar at the Media Lab, Massachusetts Institute of Technology in 2017. His research interests include mobile computing, data mining, and IoT systems. His research has been published in top-tier journals such as IEEE TMC, ACM TOIS, ACM IMWUT, and IEEE TKDE. He was awarded conference best paper twice including one from IEEE INFOCOM 2020.