mathematics

MDPI

Article

Model Fusion from Unauthorized Clients in Federated Learning

Boyuan Li [1,*], Shengbo Chen [1,*] and Keping Yu [2]

[1] School of Computer and Information Engineering, Henan University, Kaifeng 475001, China
[2] Global Information and Telecommunication Institute, Waseda University, Tokyo 169-8050, Japan
* Correspondence: 104753200808@henu.edu.cn (B.L.); 10120125@vip.henu.edu.cn (S.C.)

Abstract: A key feature of federated learning (FL) is that not all clients participate in every communication epoch of each global model update. The rationality for such partial client selection is largely to reduce the communication overhead. However, in many cases, the unselected clients are still able to compute their local model updates, but are not "authorized" to upload the updates in this round, which is a waste of computation capacity. In this work, we propose an algorithm FEDUMF—Federated Learning with Unauthorized Model Fusion that utilizes the model updates from the unselected clients. More specifically, a client computes the stochastic gradient descent (SGD) even if it is not selected to upload in the current communication epoch. Then, if this client is selected in the next round, it non-trivially merges the outdated SGD stored in the previous round with the current global model before it starts to compute the new local model. A rigorous convergence analysis is established for FEDUMF, which shows a faster convergence rate than the vanilla FEDAVG. Comprehensive numerical experiments on several standard classification tasks demonstrate its advantages, which corroborate the theoretical results.

Keywords: federated learning; convergence analysis; model fusion

MSC: 68W40; 68T09; 68Q25; 68Q85

Citation: Li, B.; Chen, S.; Yu, K. Model Fusion from Unauthorized Clients in Federated Learning. *Mathematics* **2022**, *10*, 3751. https://doi.org/10.3390/math10203751

Academic Editor: Florin Leon

Received: 27 August 2022
Accepted: 11 October 2022
Published: 12 October 2022

1. Introduction

In various large-scale machine learning (ML) applications, a massive amount of data is generated at the network edge nodes. Federated learning (FL) is a novel ML algorithm paradigm, which can effectively enable massive participants (or organizations) to execute ML algorithms and ensure privacy security without exchanging data. As a novel distributed ML paradigm, FL can deal with data isolation, allowing collaborators to train models together without sharing proprietary materials, breaking through data separation in technology and achieving collaborative ML. In particular, rather than transferring a plethora of data, all participants in FL smartly fit local ML models from their respective data and upload to the back-end server, where a global model is aggregated. In recent years, FL has achieved good performance on many commercial projects (e.g., Pixel 2 uses FL mode to personalize settings for users) and ML tasks (e.g., Gboard uses FL for prediction) [1].

It is well known that increasing communication efficiency is one of the most urgent bottlenecks in FL. There are two main reasons. First, the size of current deep learning models is usually very significant, e.g., millions of parameters. Second, the number of terminal devices participating the FL proliferates rapidly, e.g., a massive number of Internet-of-Things (IoT) devices. To address these problems, the FEDAVG algorithm [2] adopts *partial* client participation. In FEDAVG, the server randomly selects a certain portion of the total clients for model upload in each communication round, and the selected ("authorized") clients perform local model training and upload the trained models to the server for aggregation. This simple yet effective approach has enjoyed great empirical success, as it provides a flexible trade-off between a large amount of participating clients and significant communication overhead.

The novel idea of this paper, however, is motivated by the existing partial client participation mechanism in FEDAVG. In particular, the unselected clients stay idle in the current communication round and only the authorized clients actively compute model updates and send the new local models to the server. The key innovation of this work is that those unselected clients are still able to execute the model update even without authorization for model upload, which greatly wastes the computation power of those idle clients. On the other hand, in the existing literature, it has been shown that more clients participating in each round leads to a faster convergence rate. This paradox poses a new challenge: how to allow the clients without authorization to join in FL while keeping the communication overhead low.

In this paper, we propose a novel algorithm FEDUMF—Federated Learning with Unauthorized Model Fusion to address the aforementioned challenge. FEDUMF effectively leverages the computation capacity of the idle clients while keeping the number of selected clients unchanged to maintain low communication overhead. More specifically, in each round, rather than staying idle, the unselected clients also train the model in parallel with the selected counterparts. The resulting model update by running, e.g., stochastic gradient descent (SGD), will be temporarily stored locally at the clients and does not upload to the server. If any unselected client receives authorization in the next round, the stored SGD during the previous round will be fused with the global model before the new SGD is calculated in a non-trivial way that is one of the key novelties of this paper. On the other hand, if the client fails to be selected again in the next round, the previous stored SGD will be overwritten with the new SGD computed from the new received global model. Note that in this paper, we assume that all clients can receive the global model broadcast by the server in each round. This is reasonable in a wireless broadcast scenario. In some studies, it is assumed that only partial clients can receive broadcast from the server, which is different from the setting in this paper. As we can see, the FEDUMF algorithm leverages the computation capacity of the unselected clients without requiring increasing the selected clients, which may also accelerate the convergence of FL. To the best of our knowledge, this work is the first that exploits the "idle" clients without authorization in each round to boost FL performance without increasing the communication overhead.

The main contributions of this paper are summarized as follows.

1. We propose FEDUMF, which enables clients that are not authorized to upload model in each round to compute model updates and fuse the update later when they receive the upload authorization.
2. We rigorously prove the convergence behavior of FEDUMF for both strongly convex and non-convex loss functions. The results show that FEDUMF strictly dominates the standard FEDAVG.
3. We selected the standard MNIST and CIFAR10 data set in the experiment, and carried out standardized preprocessing. Extensive experiments were conducted on two common data distributions (both independent and identically distributed (IID) and non-IID data distributions).
4. We combine FEDUMF with the state-of-art algorithm to show good test accuracy and the fastest convergence speed.

This paper will be described in the following sections. Related works are surveyed in Section 2. The FEDUMF algorithm is described in Section 3 and rigorously analyzed in Section 4. Numerical experiments are presented in Section 5. Finally, Section 6 concludes the paper.

2. Related Works

Federated learning has gained much attention since its seminal work [2,3]. The proposed FEDAVG design in [2] periodically aggregates the local SGD [4] updates from massive clients with possibly non-IID datasets and obtains the global model. Detailed surveys of FL can be found in [5–7].

In [8–13], quantized compression is applied from multiple perspectives to improve the communication efficiency of FL. Ref. [14] proposes an algorithm with periodic averaging using quantization. An efficient FL protocol involving a hierarchical aggregation mechanism in a local area network (LAN) was proposed in [15], because it has abundant bandwidth and almost negligible monetary cost over WAN. Ref. [16] proposes an algorithm that achieves a compromise between gradient accuracy and communication efficiency. In [17], a combination of random rotation, subsampling, and quantization is used to decrease the transmission bandwidth. The authors in [18] applied quantization for both gradients and model parameters, and [19] developed a new algorithm that outperforms FedAvg in heterogeneous datasets. Ref. [20] proposed QSGD to achieve a compromise between convergence time and communication cost. The authors in [21] derive the convergence rate by using gradient quantization for non-convex optimization problems. Ref. [22] proposes a novel mechanism, which uses heterogeneous models to optimize the training. Ref. [23] designs an algorithm to efficiently solve the trade-off between the number of clients and transmission energy.

Another line of research that is related to this work is the study of asynchronous FL, that is, using out-of-date SGD. Synchronous training can make the model update more orderly and accurate, but the number of clients participating in the training is limited. Asynchronous training usually allows more clients to participate in model training [24,25] than synchronous training, but the cost is that the SGD used might be very dated, which may deteriorate the performance. Ref. [26] proposes a novel asynchronous adjustment method for SGD, which adjusts the learning rate according to the staleness of the offline time of the gradient, and provides theoretical guarantees for the convergence of the algorithm. In order to adapt to the access of more edge devices and improve flexibility and scalability, a novel joint optimization algorithm supporting heterogeneous terminals was proposed by [27]. Ref. [28] mitigates the straggler problem caused by device heterogeneity. Ref. [29] proposes a novel technology to compensate delay caused by asynchronous learning. However, in the previous literature, most research in FL have not considered utilizing the unselected clients to enhance the performance, while this work aims to take advantage of these neglected clients, which may bring extra benefits. In our paper, we also utilize the model updated from the unselected clients in the previous round. However, the update is not staler than one round, and the whole system is still synchronous.

3. The FEDUMF Algorithm

In this section, we introduce the federated learning system in detail and propose our algorithm named FEDUMF.

3.1. The FL Model and FEDAVG

We aim to learn a global model parameterized by θ that minimizes the following learning problem.

$$\min_{\theta \in \mathbb{R}^d} \ell(\theta) = \min_{\theta \in \mathbb{R}^d} \frac{1}{m} \sum_{z \in \mathcal{D}} l(\theta; z), \tag{1}$$

where $\ell : \mathbb{R}^d \to \mathbb{R}$ is the differentiable loss function averaged over the total dataset $\mathcal{D}$ with size m, $\theta \in \mathbb{R}^d$ is the variable of machine learning model (parameter vector) that we would like to optimize, and $l(\theta; z)$ is the loss function evaluated at data sample z and model θ. We assume that there are n clients in the FL setting. The problem can be redefined as

$$\min_{\theta \in \mathbb{R}^d} \ell(\theta) = \min_{\theta \in \mathbb{R}^d} \sum_{i \in [n]} \frac{m_i}{m} \ell_i(\theta), \tag{2}$$

where $\ell_i : \mathbb{R}^d \to \mathbb{R}$ is the local loss function for client i, averaged over its local dataset $\mathcal{D}_i$ with m_i data samples, and $\sum_i m_i = m$, i.e., $\ell_i(\theta) = \frac{1}{m_i} \sum_{z \in \mathcal{D}_i} l(\theta; z)$. Due to the space limitation, we only present the analysis for equal dataset size. Our solution can be easily

extended to the case when m_i are heterogeneous. An illustration of the considered system is given in Figure 1.

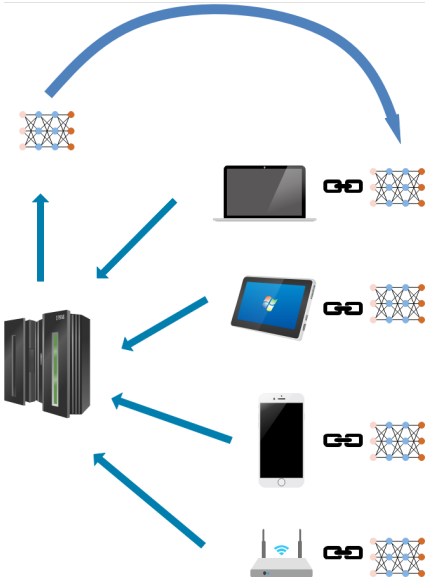

Figure 1. Illustration of federated learning. The server sends the model to all clients through broadcast and the clients submit it to the server for aggregation after local training.

In order to deal with Problem (1), the authors put forward FEDAVG in [2]. At the start of every epoch $t \in [T] \triangleq \{1, 2, \cdots, T\}$, the server broadcasts the current global model θ_t to all clients. Then, the server chooses a subset of clients S_t at random, with these clients authorized to upload their models. Assume that $|S_t| = M$; that is, a total of M clients are selected in each epoch. Then, each client $i \in S_t$ updates its local model by using K times of local SGD, obtaining θ_{t+1}^i. After the model update stage, the authorized clients upload their updates to the parameter server through uplink communication. Then, the server aggregates local model updates from selected clients to update the global model; i.e., $\theta_{t+1} = \frac{1}{M} \sum_{i \in [S_t]} \theta_{t+1}^i$. After going through T rounds of learning process, the final global model is obtained on the server.

3.2. FEDUMF *Algorithm Description*

We formally describe the FEDUMF algorithm in Algorithm 1. We denote g_t^i as the stored SGD for client i at round t, The number of all clients participating in training is N. First, we initialize a global weight θ_1 for the server and distribute it to all clients in the first round. At the same time, an update gradient $g_0^i = 0$ is initialized for all clients. At the start of each round t, the server randomly chooses an authorization set S_t, where clients are allowed to upload their model updates. We adopt S_t^c to denote the complement set of S_t, i.e., the set of unselected clients. As all the clients can obtain the global model broadcast by the server in the beginning of every round, we denote $\theta_{t,0}^i \leftarrow \theta_t$. In addition, if any selected client i at this round was not selected in the previous round, i.e., $i \in S_t \cap S_{t-1}^c$, gradient fusion from the previous stored SGD is executed $\theta_{t,0}^i \leftarrow \theta_{t,0}^i + \frac{\alpha \eta_t}{\eta_{t-1}} g_{t-1}^i$. $\alpha \in (0, 1]$ is a fusion coefficient, which is a tuning parameter. Multiplying $\frac{\eta_t}{\eta_{t-1}}$ modulates the learning rate. Notice that if the authorized client was also selected in the previous round, it indicates that the client has already contributed its previous gradient and there is no need to fuse its gradient. Then, each client $i \in S_t$ updates its local model by using K times of local SGD, obtaining θ_{t+1}^i. All clients compute and store the gradient for the current round by $g_t^i = \theta_{t+1}^i - \theta_{t,0}^i$. After receiving the model updates uploaded from all authorized clients, the server aggregates the global model by $\theta_{t+1} \leftarrow \frac{1}{M} \sum_{i \in S_t} \theta_{t+1}^i$.

Figure 2 displays the algorithm workflow. It is clear that FEDUMF is novel because of the fusion of previous SGD from unselected clients in the model update process.

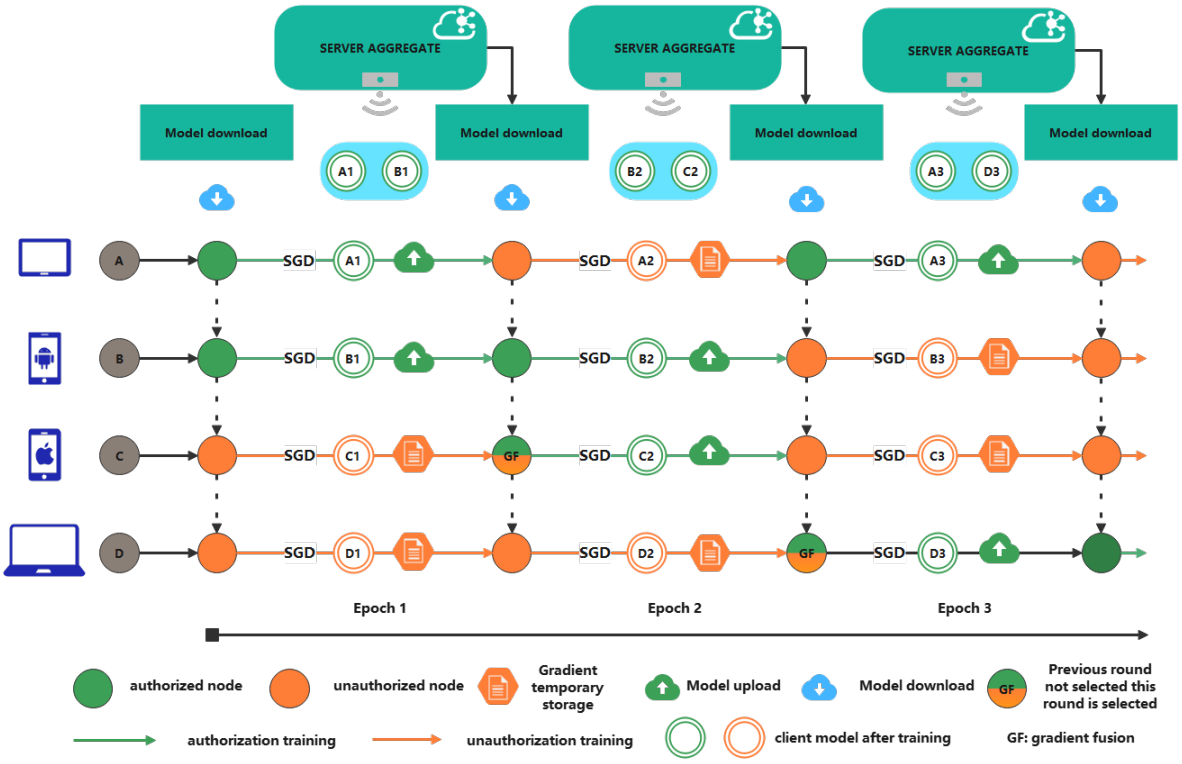

Figure 2. Illustration of FEDUMF.

Algorithm 1: FEDUMF

Initialize the weight θ_1 ;
Initialize the gradient $g_0^i = 0, \forall i$;
$S_0 = \varnothing$;
for $t = 1$ **to** T **do**
  Server randomly selects S_t ;
  for *client* $i \in N$ **do**
    $\theta_{t,0}^i \leftarrow \theta_t$;
    if *client* $i \in S_t \cap S_{t-1}^c$ **then**
      $\theta_{t,0}^i \leftarrow \theta_{t,0}^i + \frac{\alpha\eta_t}{\eta_{t-1}} g_{t-1}^i$;
    end
    for $\tau = 0$ **to** $K-1$ **do**
      $\theta_{t,\tau+1}^i = \theta_{t,\tau}^i - \eta_t \tilde{\nabla} \ell_i(\theta_{t,\tau}^i)$;
    end
    $\theta_{t+1}^i \leftarrow \theta_{t,K}^i$;
    $g_t^i = \theta_{t+1}^i - \theta_{t,0}^i$;
    if *client* $i \in S_t$ **then**
      Upload θ_{t+1}^i to the server;
    end
  end
  Server updates global model $\theta_{t+1} \leftarrow \frac{1}{M} \sum_{i \in S_t} \theta_{t+1}^i$
end

4. Analysis of Convergence

In this section, we display the convergence performance analysis result of FEDUMF. For the purpose of better illustrating of our key idea, we assume one local SGD occurs at

each round, i.e., $K = 1$, while in the experiment section, we simulate multiple local SGD updates scenarios. We also consider both strongly convex and non-convex loss functions.

4.1. Strongly Convex Loss Function

The optimal solution to Problem (2) is denoted by θ^*. We first claim some commonly adopted assumptions used in literature, e.g., [14,21,30]. Specifically, $\forall i \in [n]$. We make the following assumptions.

Assumption 1. *Each function ℓ_i is L-smooth: $\|\tilde{\nabla}\ell_i(\theta) - \tilde{\nabla}\ell_i(\hat{\theta})\| \leq L\|\theta - \hat{\theta}\|$ for any $\theta, \hat{\theta} \in \mathbb{R}^d$;*

Assumption 2. *Each function ℓ_i is μ-strongly convex: $\langle \tilde{\nabla}\ell_i(\theta) - \tilde{\nabla}\ell_i(\hat{\theta}), \theta - \hat{\theta}\rangle \geq \mu\|\theta - \hat{\theta}\|^2$ for any $\theta, \hat{\theta} \in \mathbb{R}^d$;*

Assumption 3. *The second moment of a stochastic gradient for all function ℓ_i is bounded: $\mathbb{E}\|\tilde{\nabla}\ell_i(\theta)\|^2 \leq \sigma^2$.*

Assumption 1 indicates that the gradient of ℓ_i is Lipschitz continuous. In this section, Assumption 2 assumes that the loss function is strongly convex. However, this assumption no longer holds for the analysis of the non-convex situation in the next section. Assumption 3 implies that the variance of stochastic gradients is uniformly bounded [31].

We first present three lemmas, which will be used to prove Theorem 1 later.

Lemma 1. *Under the conditions that Assumptions 1 to 3 hold, we have $\mathbb{E}\|\theta_{t+1} - \theta^*\|^2 = \mathbb{E}\|\bar{\theta}_{t+1} - \theta^*\|^2 + \mathbb{E}\|\theta_{t+1} - \bar{\theta}_{t+1}\|^2$, where $\bar{\theta}_{t+1} = \sum_{i\in[n]} \frac{1}{n}\theta^i_{t+1}$.*

Proof. First we have

$$\begin{aligned}
&\mathbb{E}\|\theta_{t+1} - \theta^*\|^2 = \mathbb{E}\|\theta_{t+1} - \bar{\theta}_{t+1} + \bar{\theta}_{t+1} - \theta^*\|^2 \\
&= \mathbb{E}\|\bar{\theta}_{t+1} - \theta^*\|^2 + \mathbb{E}\|\theta_{t+1} - \bar{\theta}_{t+1}\|^2 \\
&+ 2\mathbb{E}\langle \theta_{t+1} - \bar{\theta}_{t+1}, \bar{\theta}_{t+1} - \theta^*\rangle. \qquad (3)
\end{aligned}$$

Notice that

$$\mathbb{E}[\theta_{t+1}] = \mathbb{E}\left[\sum_{i\in S_t} \frac{1}{M}\theta^i_{t+1}\right] \overset{(a)}{=} \sum_{i\in[n]} \frac{1}{n}\theta^i_{t+1} = \bar{\theta}_{t+1}, \qquad (4)$$

where the equality (a) is because of the randomness of S_t.

Noticing that the random sampling error $\theta_{t+1} - \bar{\theta}_{t+1}$ is independent of $\bar{\theta}_{t+1} - \theta^*$, which leads to the desired result. □

Lemma 2. *When Assumptions 1 to 3 hold, we have*

$$\mathbb{E}\|\theta_{t+1} - \bar{\theta}_{t+1}\|^2 \leq \frac{(n-M)(n+3M)\eta_t^2\sigma^2}{M(n-1)n}. \qquad (5)$$

Proof. First, we can write:

$$\begin{aligned}
&\mathbb{E}_{S_t}\|\theta_{t+1} - \bar{\theta}_{t+1}\|^2 = \mathbb{E}_{S_t}\|\sum_{i\in S_t} \frac{1}{M}\theta^i_{t+1} - \bar{\theta}_{t+1}\|^2 \\
&= \mathbb{E}_{S_t}\|\frac{1}{M}(\sum_{i\in S_t} \theta^i_{t+1} - \sum_{i\in S_t} \bar{\theta}_{t+1})\|^2 \\
&= \frac{1}{M^2}\mathbb{E}_{S_t}\|\sum_{i\in[n]} \mathbb{1}(i \in S_t)(\theta^i_{t+1} - \bar{\theta}_{t+1})\|^2
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{M^2}\Big[\sum_{i\in[n]} Pr(i\in S_t)\|(\theta_{t+1}^i-\bar{\theta}_{t+1})\|^2+\sum_{i\neq j} Pr(i,j\in S_t)\langle\theta_{t+1}^i-\bar{\theta}_{t+1},\theta_{t+1}^j-\bar{\theta}_{t+1}\rangle\Big]\\
&= \frac{1}{nM}\Big[\sum_{i\in[n]}\|(\theta_{t+1}^i-\bar{\theta}_{t+1})\|^2+\frac{M-1}{nM(n-1)}\sum_{i\neq j}\langle\theta_{t+1}^i-\bar{\theta}_{t+1},\theta_{t+1}^j-\bar{\theta}_{t+1}\rangle\Big]\\
&= \frac{1}{M(n-1)}(1-\frac{M}{n})\sum_{i\in[n]}\|(\theta_{t+1}^i-\bar{\theta}_{t+1})\|^2, \qquad (6)
\end{aligned}
$$

where we adopt the equation that

$$\sum_{i\in[n]}\|(\theta_{t+1}^i-\bar{\theta}_{t+1})\|^2+\sum_{i\neq j}\langle\theta_{t+1}^i-\bar{\theta}_{t+1},\theta_{t+1}^j-\bar{\theta}_{t+1}\rangle=0.$$

On the other hand, the following holds

$$
\begin{aligned}
&\sum_{i\in[n]}\mathbb{E}\|(\theta_{t+1}^i-\bar{\theta}_{t+1})\|^2\\
&=\sum_{i\in[n]}\mathbb{E}\|(\theta_{t+1}^i-\theta_t)-(\bar{\theta}_{t+1}-\theta_t)\|^2\\
&\leq\sum_{i\in[n]}\mathbb{E}\|(\theta_{t+1}^i-\theta_t)\|^2, \qquad (7)
\end{aligned}
$$

where the last inequality holds due to $\mathbb{E}\|\theta-\mathbb{E}\theta\|^2\leq\mathbb{E}\|\theta\|^2$.

There are three cases for θ_{t+1}^i.

(1). For client $i\in S_t\cap S_{t-1}$, we have

$$\theta_{t+1}^i=\theta_t-\eta_t\tilde{\nabla}\ell_i(\theta_t).$$

(2). For client $i\in S_t\cap S_{t-1}^C$, we have

$$\theta_{t+1}^i=(\theta_t-\eta_t\tilde{\nabla}\ell_i(\theta_{t-1}))-\eta_t\tilde{\nabla}\ell_i(\theta_t-\eta_t\tilde{\nabla}\ell_i(\theta_{t-1})).$$

(3). For the other client $i\in S_t^C$, we have

$$\theta_{t+1}^i=\theta_t-\eta_t\tilde{\nabla}\ell_i(\theta_t).$$

Thus, it follows

$$
\begin{aligned}
&\sum_{i\in[n]}\|(\theta_{t+1}^i-\theta_t)\|^2\\
&=\sum_{i\in(S_t\cap S_{t-1})}\|\theta_t-\eta_t\tilde{\nabla}\ell_i(\theta_t)-\theta_t\|+\sum_{i\in S_t\cap S_{t-1}^C}\|\theta_t-\eta_t\tilde{\nabla}\ell_i(\theta_{t-1})-\eta_t\tilde{\nabla}\ell_i(\theta_t-\eta_t\tilde{\nabla}\ell_i(\theta_{t-1})-\theta_t\|\\
&+\sum_{i\in S_t\cap S_{t-1}^C}\|\theta_t-\eta_t\tilde{\nabla}\ell_i(\theta_t)-\theta_t\|\\
&=\sum_{i\in(S_t\cap S_{t-1})\cup S_t^C}\|\eta_t\tilde{\nabla}\ell_i(\theta_t)\|^2+\sum_{i\in S_t\cap S_{t-1}^C}\|\eta_t\tilde{\nabla}\ell_i(\theta_{t-1})+\eta_t\tilde{\nabla}\ell_i(\theta_t-\eta_t\tilde{\nabla}\ell_i(\theta_{t-1}))\|^2\\
&\leq\eta_t^2\Big[\sum_{i\in(S_t\cap S_{t-1})\cup S_t^C}\|\tilde{\nabla}\ell_i(\theta_t)\|^2+2\sum_{i\in S_t\cap S_{t-1}^C}\|\tilde{\nabla}\ell_i(\theta_{t-1})\|^2+2\sum_{i\in S_t\cap S_{t-1}^C}\|\tilde{\nabla}\ell_i(\theta_t-\eta_t\tilde{\nabla}\ell_i(\theta_{t-1}))\|^2\Big]. \qquad (8)
\end{aligned}
$$

The reason for the last inequality is that $\|a+b\|^2\leq 2\|a\|^2+2\|b\|^2$.

Plugging Equation (8) into (6) and taking the expectation on both sides yields

$$\mathbb{E}\|\theta_{t+1}-\bar{\theta}_{t+1}\|^2\leq\frac{(n-M)(n+3M)\eta_t^2\sigma^2}{M(n-1)n}, \qquad (9)$$

where we have used the fact that $|S_t \cap S_{t-1}^C| \leq M$ and Assumption 3. □

Lemma 3. *When Assumptions 1 to 3 hold, we have*

$$\begin{aligned}
&\mathbb{E}\|\bar{\theta}_{t+1} - \theta^*\|^2 \leq \\
&(1 - 2\mu\eta_t - \frac{2M(n-M)\mu\eta_t}{n^2} + \frac{2\eta_t^2 M(n-M)}{n^3})\mathbb{E}\|\theta_t - \theta^*\|^2 \\
&+ \frac{4M(n-M)L^2\eta_{t-1}^2\sigma^2}{n} + \frac{L^2\eta_t^2 M(n-M)\sigma^2}{n} \\
&+ \frac{\eta_t^2(n+M)^2\sigma^2}{n^2}.
\end{aligned} \tag{10}$$

Proof. First of all, we can know that

$$\begin{aligned}
&\|\bar{\theta}_{t+1} - \theta^*\|^2 = \|\frac{1}{n}\sum_{i\in[n]}\theta_{t+1}^i - \theta^*\|^2 \\
&= \|\frac{1}{n}(\sum_{i\in S_t\cap S_{t-1}}\theta_{t+1}^i + \sum_{i\in S_t\cap S_{t-1}^C}\theta_{t+1}^i + \sum_{i\in S_t^C}\theta_{t+1}^i) - \theta^*\|^2 \\
&= \|\frac{1}{n}(\sum_{i\in(S_t\cap S_{t-1})\cup S_t^C}(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_t)) + \sum_{i\in S_t\cap S_{t-1}^C}((\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1})) - \eta_t\nabla\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1}))) - \theta^*\|^2 \\
&= \|\theta_t - \frac{1}{n}\sum_{i\in[n]}\eta_t\tilde{\nabla}\ell_i(\theta_t) - \frac{1}{n}\sum_{i\in S_t\cap S_{t-1}^C}\eta_t\tilde{\nabla}\ell_i(\theta_{t-1}) + \frac{1}{n}\sum_{i\in S_t\cap S_{t-1}^C}\eta_t(\tilde{\nabla}\ell_i(\theta_t) - \tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1}))) - \theta^*\|^2 \\
&= \|\theta_t - \theta^*\|^2 - 2\langle\theta_t - \theta^*, \frac{1}{n}\sum_{i\in[n]}\eta_t\tilde{\nabla}\ell_i(\theta_t)\rangle \\
&- 2\langle\theta_t - \theta^*, \frac{1}{n}\sum_{i\in S_t\cap S_{t-1}^C}\eta_t\tilde{\nabla}\ell_i(\theta_{t-1})\rangle + 2\langle\theta_t - \theta^*, \frac{1}{n}\sum_{i\in S_t\cap S_{t-1}^C}\eta_t(\tilde{\nabla}\ell_i(\theta_t) - \tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1})))\rangle \\
&+ \|\frac{1}{n}\sum_{i\in(S_t\cap S_{t-1})\cup S_t^C}\eta_t\tilde{\nabla}\ell_i(\theta_t) + \frac{1}{n}\sum_{i\in S_t\cap S_{t-1}^C}\eta_t\tilde{\nabla}\ell_i(\theta_{t-1}) + \frac{1}{n}\sum_{i\in S_t\cap S_{t-1}^C}\eta_t\tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1}))\|^2.
\end{aligned} \tag{11}$$

We will investigate the expectation of each term in the right hand side of Equation (11). In regard to the second term, we can obtain that

$$-2\mathbb{E}\langle\theta_t - \theta^*, \frac{1}{n}\sum_{i\in[n]}\eta_t\tilde{\nabla}\ell_i(\theta_t)\rangle \leq -2\mu\eta_t\mathbb{E}\|\theta_t - \theta^*\|^2, \tag{12}$$

which is due to Assumption 2.

For the third term in the right hand side of Equation (11), we take the expectation, which yields

$$\begin{aligned}
&2\mathbb{E}\langle\theta_t - \theta^*, \frac{1}{n}\sum_{i\in S_t\cap S_{t-1}^C}\eta_t\tilde{\nabla}\ell_i(\theta_{t-1})\rangle \\
&= \frac{2\eta_t}{n}\mathbb{E}\langle\theta_t - \theta^*, \sum_{i\in S_t\cap S_{t-1}^C}\tilde{\nabla}\ell_i(\theta_t)\rangle + \frac{2\eta_t}{n}\mathbb{E}\langle\theta_t - \theta^*, \sum_{i\in S_t\cap S_{t-1}^C}(\tilde{\nabla}\ell_i(\theta_{t-1}) - \tilde{\nabla}\ell_i(\theta_t))\rangle \\
&\overset{(a)}{=} \frac{2\eta_t}{n}\mathbb{E}\langle\theta_t - \theta^*, \frac{M(n-M)}{n^2}\sum_{i\in[n]}\tilde{\nabla}\ell_i(\theta_t)\rangle + \frac{2\eta_t}{n}\mathbb{E}\langle\theta_t - \theta^*, \frac{M(n-M)}{n^2}\sum_{i\in[n]}(\tilde{\nabla}\ell_i(\theta_{t-1}) - \tilde{\nabla}\ell_i(\theta_t))\rangle \\
&\geq \frac{2M(n-M)\mu\eta_t}{n^2}\mathbb{E}\|\theta_t - \theta^*\|^2 - \frac{\eta_t^2 M(n-M)}{n^3}\mathbb{E}\|\theta_t - \theta^*\|^2 - \frac{M(n-M)}{n^3}\mathbb{E}\|\sum_{i\in[n]}(\tilde{\nabla}\ell_i(\theta_{t-1}) - \tilde{\nabla}\ell_i(\theta_t))\|^2,
\end{aligned} \tag{13}$$

where the equality (a) is established because of the randomness of S_t, and the reason why the last inequality holds is due to the help of the inequality $2\langle a, b\rangle \geq -\|a\|^2 - \|b\|^2$.

For the expectation of the fourth term in Equation (11), it follows

$$\begin{aligned}
&2\mathbb{E}\langle \theta_t - \theta^*, \frac{1}{n}\sum_{i\in S_t\cap S_{t-1}^C} \eta_t(\tilde{\nabla}\ell_i(\theta_t) - \tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1})))\rangle \\
&\overset{(a)}{=} 2\mathbb{E}\langle \theta_t - \theta^*, \frac{M(n-M)}{n^3}\sum_{i\in[n]} \eta_t(\tilde{\nabla}\ell_i(\theta_t) - \tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1})))\rangle \\
&\overset{(b)}{\leq} \frac{\eta_t^2 M(n-M)}{n^3}\mathbb{E}\|\theta_t - \theta^*\|^2 + \frac{M(n-M)}{n^3}\mathbb{E}\|\sum_{i\in[n]}(\tilde{\nabla}\ell_i(\theta_t) - \tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1})))\|^2 \\
&\overset{(c)}{\leq} \frac{\eta_t^2 M(n-M)}{n^3}\mathbb{E}\|\theta_t - \theta^*\|^2 + \frac{L^2\eta_t^2 M(n-M)}{n^3}\mathbb{E}\|\sum_{i\in[n]}\tilde{\nabla}\ell_i(\theta_{t-1})\|^2 \\
&\overset{(d)}{\leq} \frac{\eta_t^2 M(n-M)}{n^3}\mathbb{E}\|\theta_t - \theta^*\|^2 + \frac{L^2\eta_t^2 M(n-M)\sigma^2}{n},
\end{aligned} \tag{14}$$

where the equality (a) holds because of the randomness of S_t, the inequality (b) holds due to $2\langle a, b\rangle \leq \|a\|^2 + \|b\|^2$, and the inequalities (c) and (d) come from Assumptions 1 and 3, respectively.

Regarding the expectation of the last term in Equation (11), we have

$$\begin{aligned}
&\mathbb{E}\|\frac{1}{n}\sum_{i\in(S_t\cap S_{t-1})\cup S_t^C}\eta_t\tilde{\nabla}\ell_i(\theta_t) + \frac{1}{n}\sum_{i\in S_t\cap S_{t-1}^C}\eta_t\tilde{\nabla}\ell_i(\theta_{t-1}) + \frac{1}{n}\sum_{i\in S_t\cap S_{t-1}^C}\eta_t\tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1}))\|^2 \\
&\leq \frac{\eta_t^2(n+M)}{n^2}\mathbb{E}\big[\sum_{i\in(S_t\cap S_{t-1})\cup S_t^C}\|\tilde{\nabla}\ell_i(\theta_t)\|^2 + \sum_{i\in S_t\cap S_{t-1}^C}\|\tilde{\nabla}\ell_i(\theta_{t-1})\|^2 + \sum_{i\in S_t\cap S_{t-1}^C}\|\tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1}))\|^2\big] \\
&\leq \frac{\eta_t^2(n+M)^2\sigma^2}{n^2},
\end{aligned} \tag{15}$$

where the first inequality is because of the Cauchy–Schwarz inequality, and the last inequality holds due to Assumption 3.

Taking the expectation on both sides of Equation (11) and plugging in Equations (13)–(15) yields

$$\begin{aligned}
&\mathbb{E}\|\bar{\theta}_{t+1} - \theta^*\|^2 \\
&\leq (1 - 2\mu\eta_t - \frac{2M(n-M)\mu\eta_t}{n^2} + \frac{2\eta_t^2 M(n-M)}{n^3})\mathbb{E}\|\theta_t - \theta^*\|^2 \\
&+ \frac{M(n-M)}{n^3}\mathbb{E}\|\sum_{i\in[n]}(\tilde{\nabla}\ell_i(\theta_{t-1}) - \tilde{\nabla}\ell_i(\theta_t))\|^2 \\
&+ \frac{L^2\eta_t^2 M(n-M)\sigma^2}{n} + \frac{\eta_t^2(n+M)^2\sigma^2}{n^2}.
\end{aligned} \tag{16}$$

Next we analyze the second term in the right hand side of Equation (16).

$$\begin{aligned}
&\mathbb{E}\|\sum_{i\in[n]}(\tilde{\nabla}\ell_i(\theta_{t-1}) - \tilde{\nabla}\ell_i(\theta_t))\|^2 \\
&\leq \mathbb{E}\|\sum_{i\in[n]}(L(\theta_{t-1} - \theta_t))\|^2 \\
&= L^2n^2\mathbb{E}\|\theta_{t-1} - \theta_t\|^2 \\
&= L^2n^2\mathbb{E}\|\frac{1}{M}\sum_{i\in S_{t-1}\cap S_{t-2}}\eta_{t-1}\tilde{\nabla}\ell_i(\theta_{t-1}) + \frac{1}{M}\sum_{i\in S_{t-1}\cap S_{t-2}^C}\eta_{t-1}\tilde{\nabla}\ell_i(\theta_{t-2})
\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{M}\sum_{i\in S_{t-1}\cap S_{t-2}^{C}}\eta_{t-1}\tilde{\nabla}\ell_i(\theta_{t-1}-\eta_{t-1}\tilde{\nabla}\ell_i(\theta_{t-2}))\|^2\\
&\overset{(a)}{\leq}\frac{2L^2n^2\eta_{t-1}^2}{M}\Big[\sum_{i\in S_{t-1}\cap S_{t-2}}\mathbb{E}\|\tilde{\nabla}\ell_i(\theta_{t-1})\|^2+\sum_{i\in S_{t-1}\cap S_{t-2}^{C}}\mathbb{E}\|\tilde{\nabla}\ell_i(\theta_{t-2})\|^2\\
&+\sum_{i\in S_{t-1}\cap S_{t-2}^{C}}\mathbb{E}\|\tilde{\nabla}\ell_i(\theta_{t-1}-\eta_{t-1}\tilde{\nabla}\ell_i(\theta_{t-2}))\|^2\Big]\\
&\overset{(b)}{\leq}4L^2n^2\eta_{t-1}^2\sigma^2,
\end{aligned}
\tag{17}
$$

where the inequality (a) comes from Cauchy–Schwarz inequality and (b) is due to Assumption 3.

Therefore, we have

$$
\begin{aligned}
&\mathbb{E}\|\bar{\theta}_{t+1}-\theta^*\|^2\\
&\leq(1-2\mu\eta_t-\frac{2M(n-M)\mu\eta_t}{n^2}+\frac{2\eta_t^2M(n-M)}{n^3})\\
&*\mathbb{E}\|\theta_t-\theta^*\|^2+\frac{4M(n-M)L^2\eta_{t-1}^2\sigma^2}{n}\\
&+\frac{L^2\eta_t^2M(n-M)\sigma^2}{n}+\frac{\eta_t^2(n+M)^2\sigma^2}{n^2}.
\end{aligned}
\tag{18}
$$

□

We are now put forward the key theorem on the convergence of FEDUMF as follows.

Theorem 1. *With the condition that Assumptions 1 to 3 hold, and if the step size is chosen as* $\eta_t=\frac{1}{\mu t}$*, there exists a constant* t_0 *such that for any* $t>t_0$*, the convergence rate of* FEDUMF *satisfies:*

$$
\mathbb{E}\|\theta_t-\theta^*\|^2\leq\frac{t_0}{t}\mathbb{E}\|\theta_{t_0}-\theta^*\|^2+\frac{2C_0}{t-1}+\frac{C_1}{t},
\tag{19}
$$

where $C_0=\frac{4M(n-M)L^2\sigma^2}{n\mu^2}$, $C_1=\frac{L^2M(n-M)\sigma^2}{n\mu^2}+\frac{(n+M)^2\sigma^2}{n^2\mu^2}+\frac{(n-M)(n+3M)\sigma^2}{M(n-1)n\mu^2}$.

Proof. Through Lemmas 1, 2, and 3, we can obtain

$$
\begin{aligned}
&\mathbb{E}\|\theta_{t+1}-\theta^*\|^2\\
&\leq(1-2\mu\eta_t-\frac{2M(n-M)\mu\eta_t}{n^2}+\frac{2\eta_t^2M(n-M)}{n^3})\mathbb{E}\|\theta_t-\theta^*\|^2+\frac{4M(n-M)L^2\eta_{t-1}^2\sigma^2}{n}\\
&+\frac{L^2\eta_t^2M(n-M)\sigma^2}{n}+\frac{\eta_t^2(n+M)^2\sigma^2}{n^2}+\frac{(n-M)(n+3M)\eta_t^2\sigma^2}{M(n-1)n}.
\end{aligned}
\tag{20}
$$

Notice that when $\eta_t\leq\frac{1}{n}$, we have $\frac{2M(n-M)\mu\eta_t}{n^2}>\frac{2\eta_t^2M(n-M)}{n^3}$. We substitute the step size $\eta_t=\frac{1}{\mu t}$, and $a_t\triangleq\mathbb{E}\|\theta_t-\theta^*\|^2$. This leads to

$$
a_{t+1}\leq\left(1-\frac{2}{t}\right)a_t+\frac{C_0}{(t-1)^2}+\frac{C_1}{t^2}.
\tag{21}
$$

Now, we prove the theorem using induction. First, we notice that Equation (19) obviously holds when $t=t_0$. Next, we assume Equation (19) holds when $s>t_0$, i.e., $a_s\leq\frac{t_0}{s}a_{t_0}+\frac{2C_0}{s-1}+\frac{C_1}{s}$. We can know

$$
a_{s+1}\leq\left(1-\frac{2}{s}\right)a_s+\frac{C_0}{(s-1)^2}+\frac{C_1}{(s)^2}
$$

$$\begin{aligned}
&\leq \left(1-\frac{2}{s}\right)\left(\frac{t_0}{s}a_{t_0}+\frac{2C_0}{s-1}+\frac{C_1}{s}\right)+\frac{C_0}{(s-1)^2}+\frac{C_1}{s^2}\\
&=\frac{s-2}{s^2}t_0a_{t_0}+\frac{2s^2-5s+4}{s(s-1)^2}C_0+\frac{s-1}{s^2}C_1\\
&\leq \frac{t_0}{s+1}a_{t_0}+\frac{2C_0}{s}+\frac{C_1}{s+1},
\end{aligned} \tag{22}$$

which leads to the final result. □

Remark 1. Theorem 1 demonstrates that the convergence rate of FEDUMF is $O(1/T)$, which is as same as the conventional parallel SGD [31].

Remark 2. In Equation (19), the term $\frac{2M(n-M)\mu\eta_t}{n^2}-\frac{2\eta_t^2M(n-M)}{n^3}$ represents the extra benefit brought by FEDUMF, i.e., a greater decay rate for the distance between the current model and the optimum than the counterpart in FEDAVG, which implies a faster convergence rate.

4.2. Non-Convex Loss Function

Neural networks are often trained using non-convex loss functions. We now pay attention to the convergence performance for non-convex loss functions; i.e., we will not consider Assumption 2 in the analysis. Specially, it is well-known that SGD may converge to a local optimum for a non-convex loss function, and evaluating the expected gradient norm as an indicator of convergence is a common practice. In particular, an algorithm achieves an ϵ-suboptimal solution if

$$\sum_{t=1}^{T}\mathbb{E}\|\nabla\ell(\theta_t)\|^2\leq\epsilon,$$

which guarantees the convergence to a stationary point [11,32,33].

Theorem 2. *Suppose Assumptions 1 and 3 are established. When the step size is substituted as* $\eta_t=\frac{1}{L\sqrt{T}}$, *the convergence of* FEDUMF *with non-convex loss functions satisfies*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla\ell(\theta_t)\|^2\leq\frac{2L(\ell(\theta_0)-f^*)}{\sqrt{T}}+\frac{D}{\sqrt{T}}, \tag{23}$$

where $D=\frac{2L\sigma^2M(n-M)}{n}+\frac{LM(n-M)\sigma^2}{2n}+\frac{(n+M)^2\sigma^2}{2n^2}+\frac{(n-M)(n+3M)\sigma^2}{2M(n-1)n}$.

We establish the following lemma before providing the full proof of Theorem 2.

Lemma 4. *When Assumption 1 holds, we can obtain*

$$\mathbb{E}\ell(\theta_{t+1})\leq\mathbb{E}\ell(\bar{\theta}_{t+1})+\frac{L}{2}\mathbb{E}\|\theta_{t+1}-\bar{\theta}_{t+1}\|^2, \tag{24}$$

$$\begin{aligned}
&\mathbb{E}\ell(\bar{\theta}_{t+1})\leq\mathbb{E}\ell(\theta_t)-\eta_t\mathbb{E}\|\nabla\ell(\theta_t)\|^2-\\
&\frac{M(n-M)\eta_t}{n^2}\mathbb{E}\|\nabla\ell(\theta_t)\|^2+\frac{\eta_t^2M(n-M)}{n^3}\mathbb{E}\|\nabla\ell(\theta_t)\|^2+\frac{2L^2\eta_{t-1}^2\sigma^2M(n-M)}{n}+\\
&\frac{L^2\eta_t^2M(n-M)\sigma^2}{2n}+\frac{\eta_t^2(n+M)^2\sigma^2L}{2n^2}.
\end{aligned} \tag{25}$$

Proof. To prove Equation (24), we note that

$$\begin{aligned}
\ell(\theta_{t+1})&=\ell(\bar{\theta}_{t+1}+\theta_{t+1}-\bar{\theta}_{t+1})\\
&\leq\ell(\bar{\theta}_{t+1})+\langle\nabla\ell(\bar{\theta}_{t+1}),\theta_{t+1}-\bar{\theta}_{t+1}\rangle+\frac{L}{2}\|\theta_{t+1}-\bar{\theta}_{t+1}\|^2,
\end{aligned} \tag{26}$$

where the inequality is established because of the L-smoothness of function ℓ. Taking expectation on both sides, we can know that $\mathbb{E}[\theta_{t+1}] = \mathbb{E}[\bar{\theta}_{t+1}]$ and the random sampling error $\theta_{t+1} - \bar{\theta}_{t+1}$ is independent of $\nabla\ell(\bar{\theta}_{t+1})$. This yields $\mathbb{E}\ell(\theta_{t+1}) \leq \mathbb{E}\ell(\bar{\theta}_{t+1}) + \frac{L}{2}\mathbb{E}\|\theta_{t+1} - \bar{\theta}_{t+1}\|^2$.

In order to prove Equation (25), we have $\ell(\bar{\theta}_{t+1}) = \ell(\frac{1}{n}\sum_{i\in[n]}\theta^i_{t+1})$

$$
\begin{aligned}
&= \ell(\frac{1}{n}\sum_{i\in S_t\cap S_{t-1}}(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_t)) + \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1})) - \eta_t\tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1}))\\
&+ \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_t))\\
&= \ell\Big(\theta_t - \frac{1}{n}\sum_{i\in(S_t\cap S_{t-1})\cup S^C_t}\eta_t\tilde{\nabla}\ell_i(\theta_t) - \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\eta_t\tilde{\nabla}\ell_i(\theta_t) + \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\eta_t\tilde{\nabla}\ell_i(\theta_t)\\
&- \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\eta_t\tilde{\nabla}\ell_i(\theta_{t-1}) - \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\eta_t(\tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1})))\Big)\\
&= \ell\Big(\theta_t - \frac{1}{n}\sum_{i\in[n]}\eta_t\tilde{\nabla}\ell_i(\theta_t) - \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\eta_t\tilde{\nabla}\ell_i(\theta_{t-1}) + \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\eta_t(\tilde{\nabla}\ell_i(\theta_t) - \tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1})))\Big)\\
&\leq \ell(\theta_t) - \langle\nabla\ell(\theta_t), \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\eta_t\tilde{\nabla}\ell_i(\theta_{t-1})\rangle\\
&- \langle\nabla\ell(\theta_t), \eta_t\nabla\ell(\theta_t)\rangle + \langle\nabla\ell(\theta_t), \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\eta_t(\tilde{\nabla}\ell_i(\theta_t) - \tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1})))\rangle +\\
&\frac{L\eta_t^2}{2}\|\frac{1}{n}\sum_{i\in(S_t\cap S_{t-1})\cup S^C_t}\tilde{\nabla}\ell_i(\theta_t) + \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\tilde{\nabla}\ell_i(\theta_{t-1}) + \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\tilde{\nabla}\ell_i(\theta_t - \eta_t\tilde{\nabla}\ell_i(\theta_{t-1}))\|^2. \qquad (27)
\end{aligned}
$$

The reason why the inequality holds relies on the L-smoothness of ℓ.
We will investigate the expectation of each term in the right hand side of Equation (27).
For the second term in the right hand side of Equation (27), we can obtain

$$
\begin{aligned}
&\mathbb{E}\langle\nabla\ell(\theta_t), \frac{1}{n}\sum_{i\in S_t\cap S^C_{t-1}}\eta_t\tilde{\nabla}\ell_i(\theta_{t-1})\rangle\\
&= \frac{\eta_t}{n}\mathbb{E}\langle\nabla\ell(\theta_t), \sum_{i\in S_t\cap S^C_{t-1}}\tilde{\nabla}\ell_i(\theta_t)\rangle\\
&+ \frac{\eta_t}{n}\mathbb{E}\langle\nabla\ell(\theta_t), \sum_{i\in S_t\cap S^C_{t-1}}(\tilde{\nabla}\ell_i(\theta_{t-1}) - \tilde{\nabla}\ell_i(\theta_t))\rangle\\
&\overset{(a)}{=} \frac{\eta_t}{n}\mathbb{E}\langle\nabla\ell(\theta_t), \frac{M(n-M)}{n^2}\sum_{i\in[n]}\tilde{\nabla}\ell_i(\theta_t)\rangle +\\
&\frac{\eta_t}{n}\mathbb{E}\langle\nabla\ell(\theta_t), \frac{M(n-M)}{n^2}\sum_{i\in[n]}(\tilde{\nabla}\ell_i(\theta_{t-1}) - \tilde{\nabla}\ell_i(\theta_t))\rangle\\
&\overset{(b)}{\geq} \frac{M(n-M)\eta_t}{n^2}\mathbb{E}\|\nabla\ell(\theta_t)\|^2 - \frac{\eta_t^2M(n-M)}{2n^3}\mathbb{E}\|\nabla\ell(\theta_t)\|^2\\
&- \frac{M(n-M)}{2n^3}\mathbb{E}\|\sum_{i\in[n]}(\tilde{\nabla}\ell_i(\theta_{t-1}) - \tilde{\nabla}\ell_i(\theta_t))\|^2\\
&\overset{(c)}{\geq} \frac{M(n-M)\eta_t}{n^2}\mathbb{E}\|\nabla\ell(\theta_t)\|^2 - \frac{\eta_t^2M(n-M)}{2n^3}\mathbb{E}\|\nabla\ell(\theta_t)\|^2\\
&- \frac{2L^2\eta_{t-1}^2\sigma^2M(n-M)}{n}, \qquad (28)
\end{aligned}
$$

where the equality (a) holds because of the randomness of S_t, the inequality (b) holds due to $2\langle a, b\rangle \geq -\|a\|^2 - \|b\|^2$, and the inequality (c) comes from Equation (17).

For the third term in the right hand side of Equation (27), we have

$$\mathbb{E}\langle \nabla \ell(\theta_t), \eta_t \nabla \ell(\theta_t)\rangle = \eta_t \mathbb{E}\|\nabla \ell(\theta_t)\|^2. \tag{29}$$

For the fourth term in the right hand side of Equation (27), we have

$$\begin{aligned}
&\mathbb{E}\langle \nabla \ell(\theta_t), \frac{1}{n} \sum_{i \in S_t \cap S_{t-1}^C} \eta_t (\tilde{\nabla} \ell_i(\theta_t) - \tilde{\nabla} \ell_i(\theta_t - \eta_t \tilde{\nabla} \ell_i(\theta_{t-1})))\rangle \\
&\overset{(a)}{=} \mathbb{E}\langle \nabla \ell(\theta_t), \frac{M(n-M)}{n^3} \sum_{i \in [n]} \eta_t (\tilde{\nabla} \ell_i(\theta_t) - \tilde{\nabla} \ell_i(\theta_t - \eta_t \tilde{\nabla} \ell_i(\theta_{t-1})))\rangle \\
&\overset{(b)}{\leq} \frac{\eta_t^2 M(n-M)}{2n^3} \mathbb{E}\|\nabla \ell(\theta_t)\|^2 + \frac{M(n-M)}{2n^3} \mathbb{E}\| \sum_{i \in [n]} (\tilde{\nabla} \ell_i(\theta_t) - \tilde{\nabla} \ell_i(\theta_t - \eta_t \tilde{\nabla} \ell_i(\theta_{t-1})))\|^2 \\
&\overset{(c)}{\leq} \frac{\eta_t^2 M(n-M)}{2n^3} \mathbb{E}\|\nabla \ell(\theta_t)\|^2 + \frac{L^2 \eta_t^2 M(n-M)}{2n^3} \mathbb{E}\| \sum_{i \in [n]} \tilde{\nabla} \ell_i(\theta_{t-1})\|^2 \\
&\overset{(d)}{\leq} \frac{\eta_t^2 M(n-M)}{2n^3} \mathbb{E}\|\nabla \ell(\theta_t)\|^2 + \frac{L^2 \eta_t^2 M(n-M)\sigma^2}{2n},
\end{aligned} \tag{30}$$

where the equality (a) holds because of the randomness of S_t, the inequality (b) holds due to $2\langle a, b\rangle \leq \|a\|^2 + \|b\|^2$, the inequality (c) comes from Assumption 1, and the inequality (d) comes from Assumption 3.

For the last term in the right hand side of Equation (27), it follows

$$\begin{aligned}
&\frac{L\eta_t^2}{2} \mathbb{E}\| \frac{1}{n} \sum_{i \in (S_t \cap S_{t-1}) \cup S_t^C} \tilde{\nabla} \ell_i(\theta_t) + \frac{1}{n} \sum_{i \in S_t \cap S_{t-1}^C} \tilde{\nabla} \ell_i(\theta_{t-1}) + \frac{1}{n} \sum_{i \in S_t \cap S_{t-1}^C} \tilde{\nabla} \ell_i(\theta_t - \eta_t \tilde{\nabla} \ell_i(\theta_{t-1}))\|^2 \\
&\leq \frac{\eta_t^2 (n+M)^2 \sigma^2 L}{2n^2},
\end{aligned} \tag{31}$$

where the last inequality comes from Equation (15).

Plugging Equations (28)–(31) into (27) and taking the expectation on both sides yields

$$\begin{aligned}
\mathbb{E}\ell(\bar{\theta}_{t+1}) \leq\ & \mathbb{E}\ell(\theta_t) - \eta_t \mathbb{E}\|\nabla \ell(\theta_t)\|^2 \\
& - \frac{M(n-M)\eta_t}{n^2} \mathbb{E}\|\nabla \ell(\theta_t)\|^2 + \frac{\eta_t^2 M(n-M)}{n^3} \mathbb{E}\|\nabla \ell(\theta_t)\|^2 \\
& + \frac{2L^2 \eta_{t-1}^2 \sigma^2 M(n-M)}{n} + \frac{L^2 \eta_t^2 M(n-M)\sigma^2}{2n} \\
& + \frac{\eta_t^2 (n+M)^2 \sigma^2 L}{2n^2}.
\end{aligned} \tag{32}$$

□

Proof of Theorem 2. From Lemmas 2 and 4, we have

$$\begin{aligned}
\mathbb{E}\ell(\theta_{t+1}) \leq\ & \mathbb{E}\ell(\theta_t) - \eta_t \mathbb{E}\|\nabla \ell(\theta_t)\|^2 - \\
& \frac{M(n-M)\eta_t}{n^2} \mathbb{E}\|\nabla \ell(\theta_t)\|^2 + \frac{\eta_t^2 M(n-M)}{n^3} \mathbb{E}\|\nabla \ell(\theta_t)\|^2 \\
& + \frac{2L^2 \eta_{t-1}^2 \sigma^2 M(n-M)}{n} + \frac{L^2 \eta_t^2 M(n-M)\sigma^2}{2n} \\
& + \frac{\eta_t^2 (n+M)^2 \sigma^2 L}{2n^2} + \frac{(n-M)(n+3M)\eta_t^2 \sigma^2 L}{2M(n-1)n}.
\end{aligned} \tag{33}$$

When the step size $\eta_t = \eta_{t-1} = \frac{1}{L\sqrt{T}}$, it follows $\frac{M(n-M)\eta_t}{n^2} \geq \frac{M(n-M)\eta_t^2}{n^3}$ and we have

$$\mathbb{E}\ell(\theta_{t+1}) \leq \mathbb{E}\ell(\theta_t) - \eta_t \mathbb{E}\|\nabla\ell(\theta_t)\|^2 + D\eta_t^2 L. \tag{34}$$

Summing Equation (34) over $t = 0, \cdots, T$ and rearranging the terms, we obtain

$$\begin{aligned}\sum_{t=0}^{T} \eta_t \mathbb{E}\|\nabla\ell(\theta_t)\|^2 &\leq \ell(\theta_0) - \mathbb{E}\ell(\theta_{T+1}) + D\eta_t^2 LT \\ &\leq \ell(\theta_0) - \ell^* + D\eta_t^2 LT. \end{aligned} \tag{35}$$

We divide T for both sides of Equation (35), which leads to

$$\frac{1}{T}\sum_{t=0}^{T} \eta_t \mathbb{E}\|\nabla\ell(\theta_t)\|^2 \leq \frac{\ell(\theta_0) - \ell^*}{T} + D\eta_t^2 L. \tag{36}$$

We can obtain the final result (23) by plugging in the step size $\eta_t = \frac{1}{L\sqrt{T}}$. □

Remark 3. Theorem 2 demonstrates that FEDUMF has a convergence rate of $O(1/\sqrt{T})$ for non-convex loss functions, which is as same as the conventional parallel SGD without quantization [33].

5. Experiments

5.1. Experimental Setup

5.1.1. Models and Datasets

In this experiment, we employ real-world datasets for standard image classification tasks, including the commonly used MNIST and CIFAR10. In Appendix A, we show more experimental results; the dataset is extended to MNIST, EMNIST, Fashion-MNIST, CIFAR10 and CIFAR100.

MNIST is a simple digit classification dataset; the training set contains 60,000 (28∗28) samples and the test set contains 10,000 samples. In addition, CIFAR10 is also intended for an image classification task with 10 classes, and its data set contains a total of 60,000 (32∗32) samples available for use. In particular, our network setup follows that described in the literature [2] on MNSIT, ResNet18 [34] on CIFAR10.

The experimental design mainly includes two mainstream balance settings: IID setting and non-IID setting. For the IID setup, we shuffle the dataset randomly; all clients receive the same number of samples, which are independently and equally distributed over the training data set. The non-IID setup has a very different setting than IID. We divide the data into blocks by category and each person extracts several blocks. According to the client settings, the client will hold categories ranging from 1 to n. In this case, we make it have at most two classes.

5.1.2. Hyperparameters

In this section, we describe in detail the meanings and settings of some of the hyperparameters used in the experiments. We set T as the communication rounds. B is the local batch size for training at each round and $C \in (0, 1]$ is the proportion of clients authorized by the server each round. Let K represent the local training times of each client. We test multiple learning rate settings. The low learning rate η is set as 0.01, while the high one is set as 0.05. In addition, the low client fraction C is set as 0.1, while the high one is set as 0.5. Mo stands for the momentum choice of the optimizer. λ stands for the weight decay of the optimizer, which is set to be 0.0005. For MNIST, the settings are $B = 50$, $T = 150$, $K = 1$, and $Mo = 0.5$. For CIFAR10, the settings are $B = 50$, $T = 300$, $K = 1$, and $Mo = 0.5$.

5.1.3. Criteria

All models are appraised at each epoch using the test accuracy, and we also record the number training rounds to arrive at a given test accuracy. The loss function is calculated using cross-entropy. In the main text, we use FedAvg as the baseline, and in the Appendix A, we compare more algorithms, such as FedProx, FedScaff, FedDyn, and FedDC (state of the art). All experiments are run using Tesla V100 with 125GB RAM. In addition to using a figure to show the test accuracy curve, we also use a table to show the communication rounds when the specified test accuracy is achieved.

5.2. MNIST Results

5.2.1. IID Setting

Table 1 records the experimental results for the MNIST dataset with the IID partition for the case of $\alpha = 1$, which means that the full SGD of the unselected clients in the previous round will be fused in this round. In the top two rows of the table, the learning rate is set as $\eta = 0.01$ and C is set to be 0.1 and 0.5, respectively. In the bottom two rows of the table, we have $\eta = 0.05$, and C is also set to be 0.1 and 0.5, respectively. We can see that the FEDUMF outperforms FEDAVG for all cases. For example, the number of rounds to reach the test accuracy of 98% is 34–48% smaller.

Table 1. Accuracy on MNIST dataset. IID partition.

Parameters, $\alpha = 1$	**Schemes**	**Accuracy**					
		92%	**93%**	**94%**	**95%**	**96%**	**97%**
$\eta = 0.01,\ C = 0.1$	FEDAVG	23	34	48	69	103	*
	FEDUMF	14	21	28	39	55	99
$\eta = 0.01,\ C = 0.5$	FEDAVG	23	35	46	70	101	*
	FEDUMF	17	24	32	47	66	110
$\eta = 0.05,\ C = 0.1$	FEDAVG	6	9	11	15	23	38
	FEDUMF	5	6	7	10	15	25
$\eta = 0.05,\ C = 0.5$	FEDAVG	4	5	6	11	15	46
	FEDUMF	4	5	6	9	15	29

'*' is used to denote that the model fails to achieve the target accuracy.

Subfigures (a) and (b) in Figures 3 and 4 plot the number of communication rounds versus test accuracy for the two aforementioned scenarios. Subfigures (c) and (d) in Figures 3 and 4 plot the number of communication rounds versus the corresponding training loss. For a given learning rate, a greater C leads to a smaller gap between FEDUMF and FEDAVG, because a larger C implies increasing the number of clients selected at each round, which results in a reduction of the number of unselected clients whose gradient can be leveraged. In addition, to illustrate the effect of the coefficient α, we also add two curves for $\alpha = 0.1$ and $\alpha = 0.5$ as comparison. As α increases, performance becomes better, which means that adding the full gradient is the best choice for the current setting. When the learning rate increases from 0.1 to 0.5, a similar performance enhancement can be observed, which indicates the wide applicable range of our algorithm.

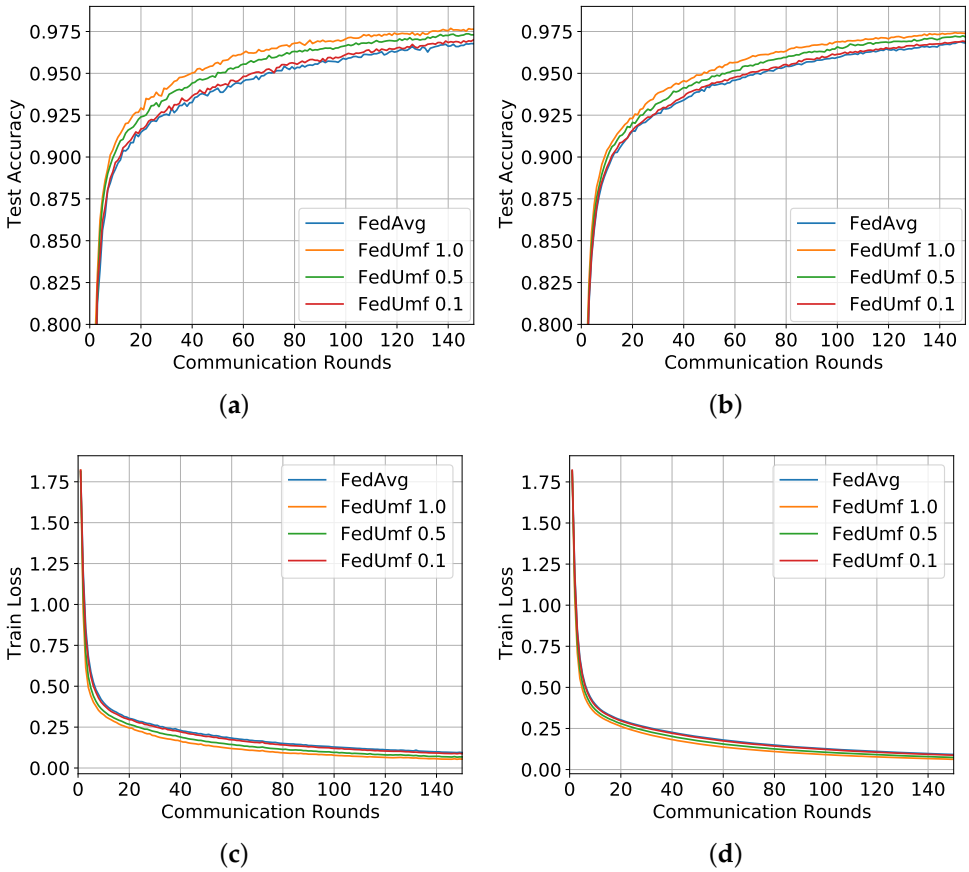

Figure 3. Server test accuracy and client-side training loss curve performance, IID partition and $\eta = 0.01$ on MNIST. (**a**) Test accuracy. $C = 0.1$. (**b**) Test accuracy. $C = 0.5$. (**c**) Training loss. $C = 0.1$. (**d**) Training loss. $C = 0.5$.

5.2.2. Non-IID Setting

Table 2 records the experimental results for the MNIST dataset with the non-IID partition for the case of $\alpha = 1$. Similar settings are adopted as in the IID case. We can see that the performance of FEDUMF dominates FEDAVG for all cases as well.

Table 2. Accuracy on MNIST dataset, Non-IID partition.

Parameters, $\alpha = 1$	**Schemes**	**Accuracy**					
		91%	**92%**	**93%**	**94%**	**95%**	**96%**
$\eta = 0.01,\ C = 0.1$	FEDAVG	24	34	44	71	92	141
	FEDUMF	12	18	29	34	54	69
$\eta = 0.01,\ C = 0.5$	FEDAVG	22	33	44	64	92	135
	FEDUMF	15	23	32	44	62	87
$\eta = 0.05,\ C = 0.1$	FEDAVG	7	8	11	14	22	31
	FEDUMF	5	7	8	10	11	23
$\eta = 0.05,\ C = 0.5$	FEDAVG	7	8	11	14	19	26
	FEDUMF	5	6	8	10	14	21

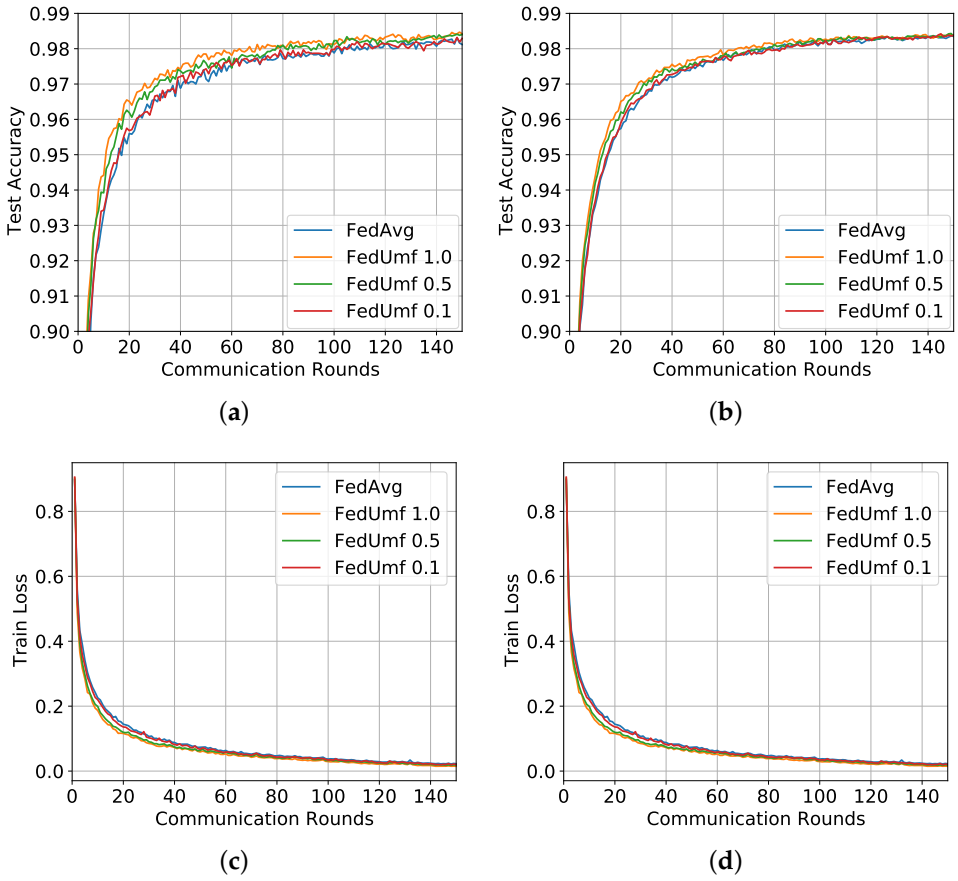

Figure 4. Server test accuracy and client-side training loss curve performance, IID partition and $\eta = 0.05$ on MNIST. (**a**) Test accuracy. $C = 0.1$. (**b**) Test accuracy. $C = 0.5$. (**c**) Training loss. $C = 0.1$. (**d**) Training loss. $C = 0.5$.

Subfigures (a) and (b) in Figures 5 and 6 plot the number of communication rounds versus test accuracy for the two aforementioned scenarios, while subfigures (c) and (d) plot the corresponding training loss. We can still observe a clear performance boost for FEDUMF compared to FEDAVG. Similarly, it can be seen that a larger C implies a smaller gap between FEDUMF and FEDAVG, for the same reason as in the IID case.

5.3. CIFAR10 Result

5.3.1. IID Setting

We adopt the ResNet18 CNN model used in [34] with $\alpha = 1$. Table 3 reports the results. '*' is used to denote that the model fails to achieve the target accuracy. In the top two rows of the table, the learning rate is set as $\eta = 0.05$ and C is set to be 0.1 and 0.5, respectively. In the bottom two rows of the table, we have $\eta = 0.1$ and C is also set to be 0.1 and 0.5, respectively. Similar to the experiment on MNIST dataset, we notice that FEDUMF outperforms FEDAVG significantly in all cases; e.g., the number of rounds to reach an accuracy of 70% is 42–50% less for all cases. When C increases, we can see that the benefits of our algorithm gradually diminish, which is because the chance of utilizing gradient fusions decreases.

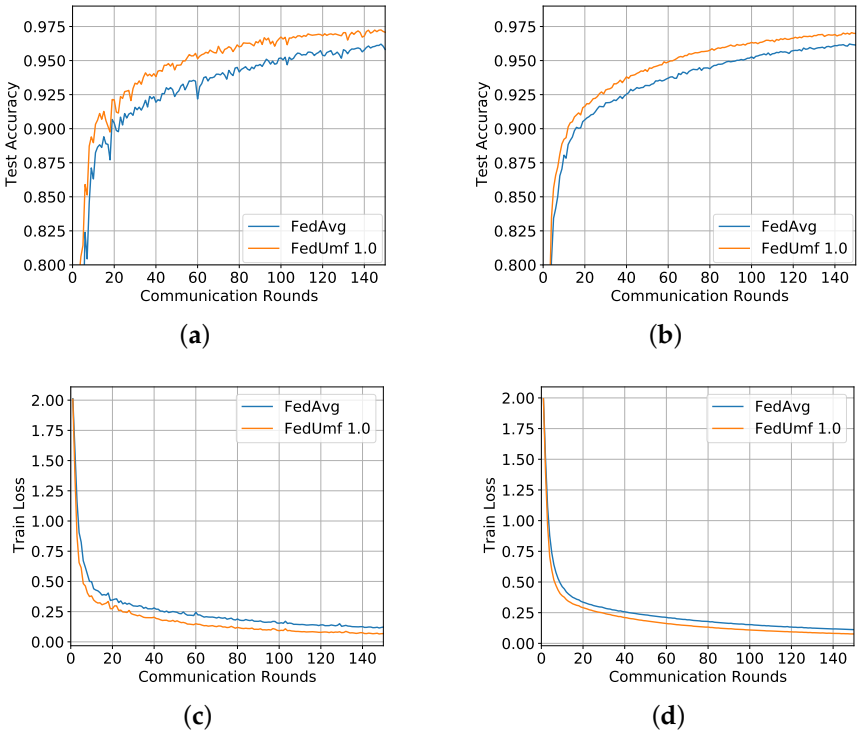

Figure 5. Server test accuracy, and client-side training loss curve performance, non-IID partition and $\eta = 0.01$ on MNIST. (**a**) Test accuracy. $C = 0.1$. (**b**) Test accuracy. $C = 0.5$. (**c**) Training loss. $C = 0.1$. (**d**) Training loss. $C = 0.5$.

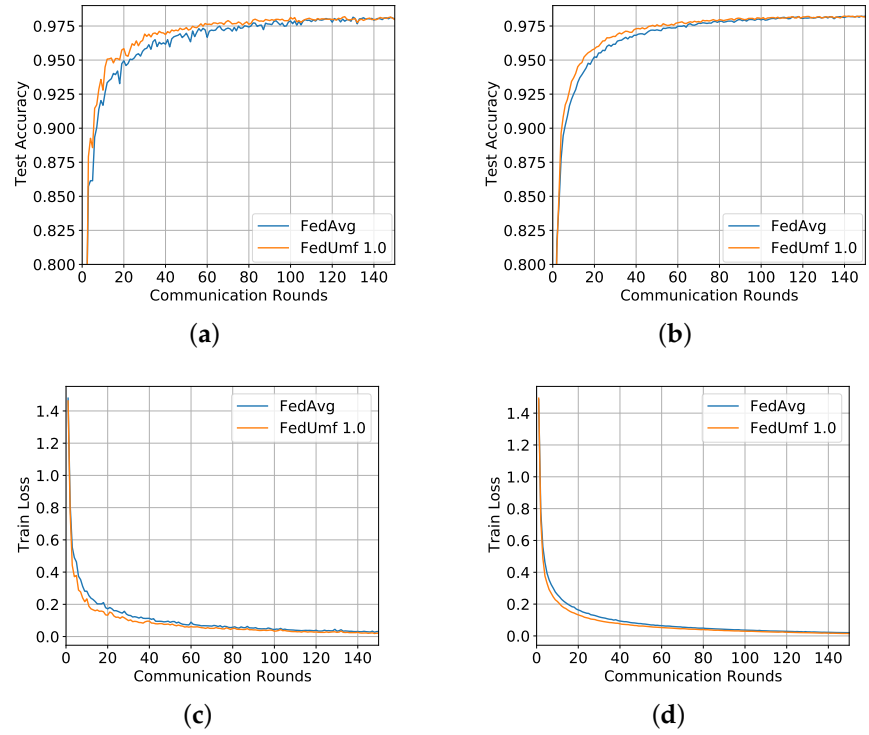

Figure 6. Server test accuracy, and client-side training loss curve performance, non-IID partition and $\eta = 0.05$ on MNIST. (**a**) Test accuracy. $C = 0.1$. (**b**) Test accuracy. $C = 0.5$. (**c**) Training loss. $C = 0.1$. (**d**) Training loss. $C = 0.5$.

Table 3. Accuracy on CIFAR10 dataset, IID partition.

Parameters, $\alpha = 1$	**Schemes**	**Accuracy**						
		30%	**40%**	**50%**	**55%**	**60%**	**65%**	**70%**
$\eta = 0.01,\ C = 0.1$	FedAvg	6	25	82	130	207	*	*
	FedUmf	4	15	45	66	103	165	288
$\eta = 0.01,\ C = 0.5$	FedAvg	4	25	78	128	200	*	*
	FedUmf	3	17	52	85	129	206	*
$\eta = 0.05,\ C = 0.1$	FedAvg	4	11	33	53	76	108	157
	FedUmf	4	8	21	31	45	64	90
$\eta = 0.05,\ C = 0.5$	FedAvg	3	10	30	46	67	97	140
	FedUmf	3	9	22	34	49	70	101

Subfigures (a) and (b) in Figures 7 and 8 plot the number of communication rounds versus the test accuracy for the two aforementioned scenarios. Subfigures (c) and (d) in Figures 7 and 8 plot the number of communication rounds versus the corresponding training loss for these two scenarios. We have also simulated $\alpha = 0.1$ and $\alpha = 0.5$ cases. The results show a descending trend as α becomes smaller, similar to the MNIST dataset.

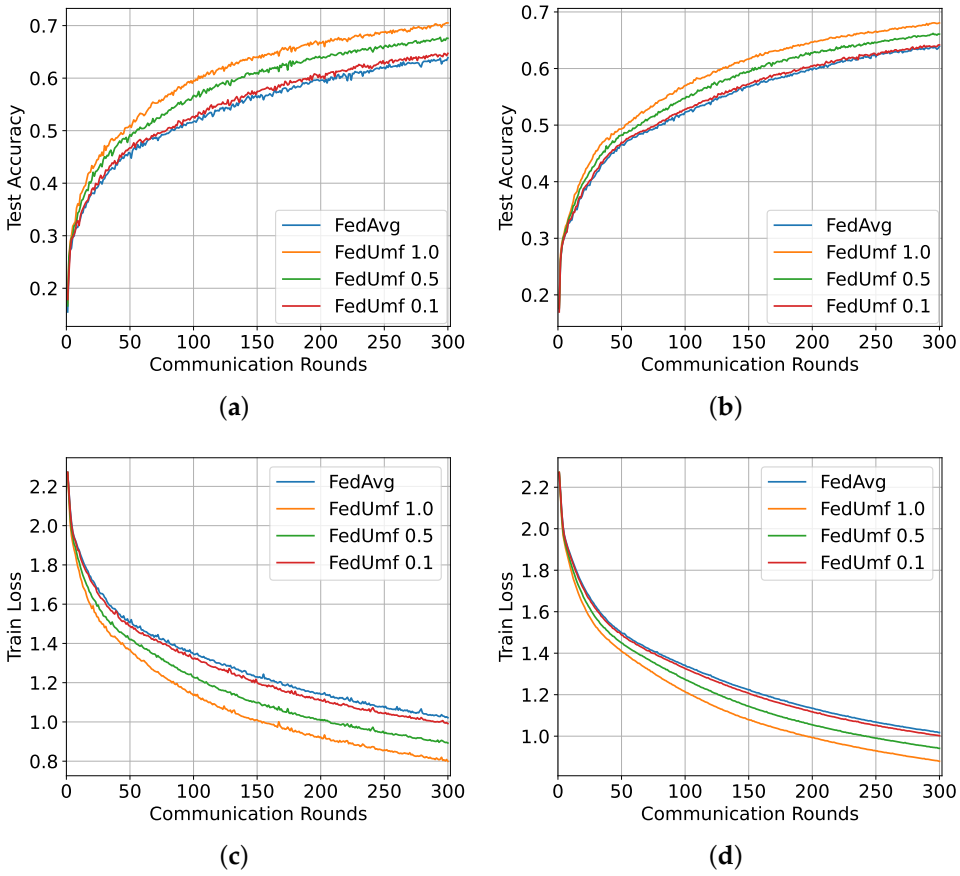

Figure 7. Server test accuracy and client-side training loss curve performance, IID partition and $\eta = 0.01$ on CIFAR. (**a**) Test accuracy. $C = 0.1$. (**b**) Test accuracy. $C = 0.5$. (**c**) Training loss. $C = 0.1$. (**d**) Training loss. $C = 0.5$.

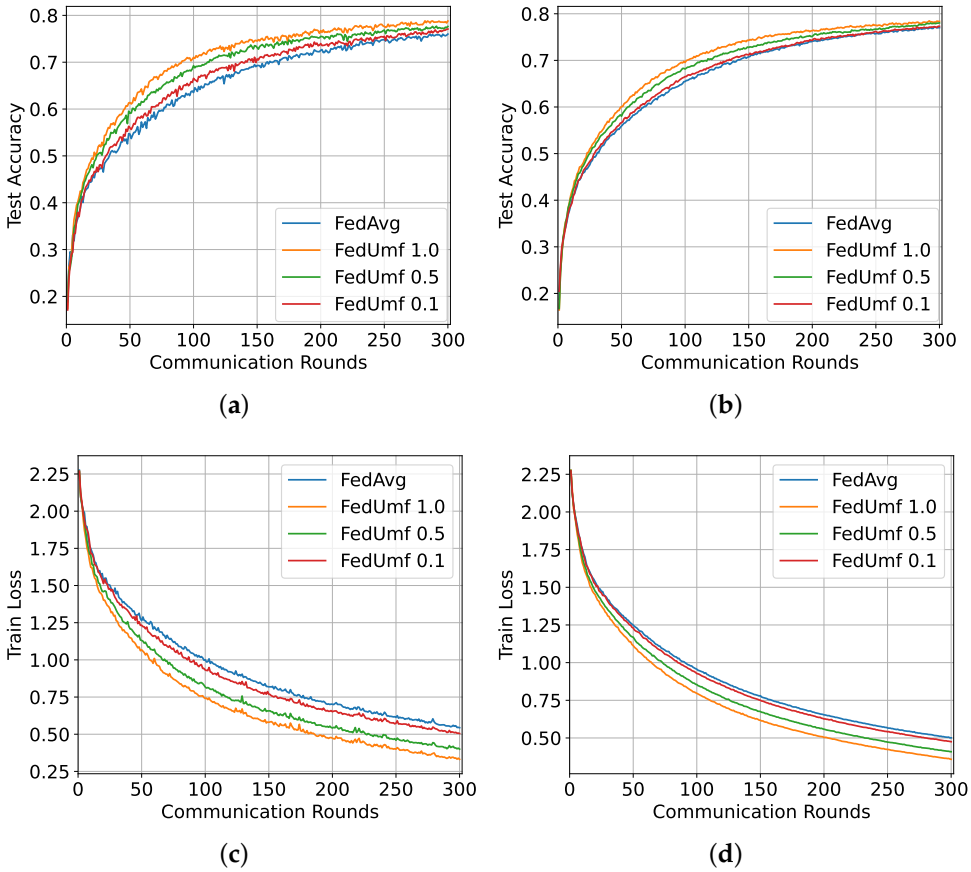

Figure 8. Server test accuracy and client-side training loss curve performance, IID partition and $\eta = 0.05$ on CIFAR. (**a**) Test accuracy. $C = 0.1$. (**b**) Test accuracy. $C = 0.5$. (**c**) Training loss. $C = 0.1$. (**d**) Training loss. $C = 0.5$.

5.3.2. Non-IID Partition

Figures 9 and 10 and Table 4 show the results for CIFAR10 with the non-IID partition. We set $\eta = 0.01$ and simulate the cases with $C = 0.1$ and $C = 0.5$. In addition, we set $\eta = 0.05$ and simulate the cases with $C = 0.1$ and $C = 0.5$. It can be seen that FEDUMF still achieves a faster convergence rate than FEDAVG.

Table 4. Accuracy on CIFAR10 dataset, non-IID partition.

Parameters, $\alpha = 0.5$	**Schemes**	**Accuracy**					
		45%	**50%**	**55%**	**60%**	**65%**	**70%**
$\eta = 0.01,\ C = 0.1$	FEDAVG	74	129	224	*	*	*
	FEDUMF	43	65	103	146	230	*
$\eta = 0.01,\ C = 0.5$	FEDAVG	69	124	194	*	*	*
	FEDUMF	44	78	122	183	290	*
$\eta = 0.05,\ C = 0.1$	FEDAVG	31	43	65	95	129	191
	FEDUMF	17	25	39	54	74	108
$\eta = 0.05,\ C = 0.5$	FEDAVG	24	37	57	82	116	167
	FEDUMF	18	26	40	57	81	116

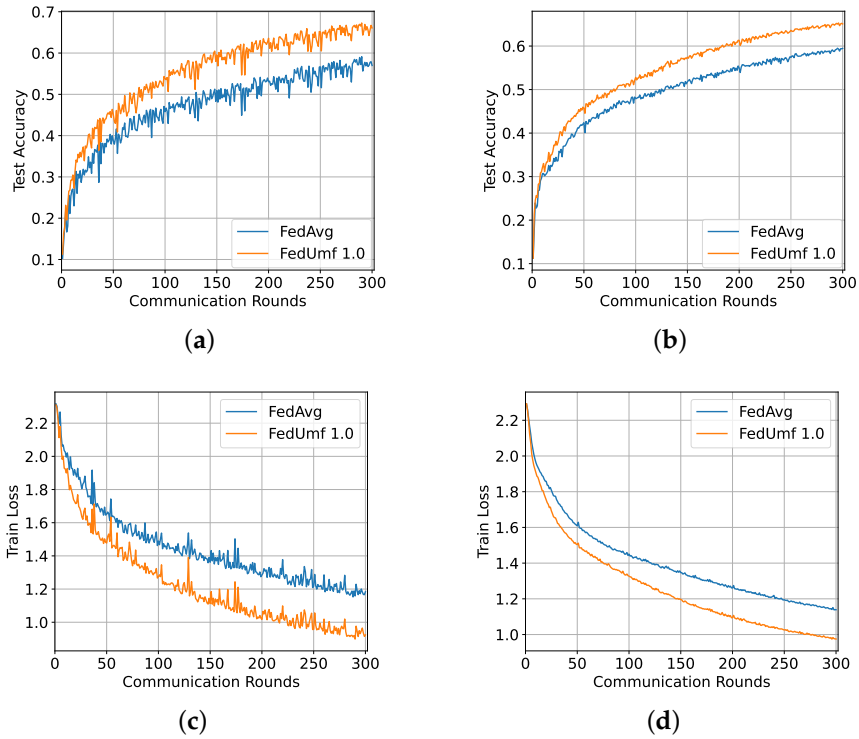

Figure 9. Server test accuracy and client-side training loss curve performance. non-IID partition, $\eta = 0.01$ on CIFAR10. (**a**) Test accuracy. $C = 0.1$. (**b**) Test accuracy. $C = 0.5$. (**c**) Training loss. $C = 0.1$. (**d**) Training loss. $C = 0.5$.

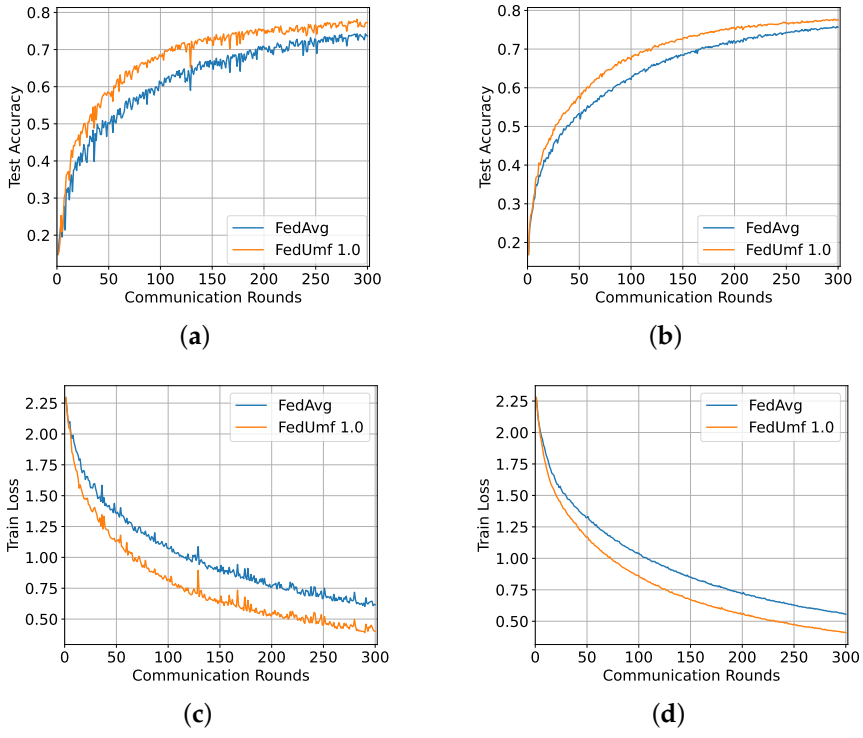

Figure 10. Server test accuracy and client-side training loss curve performance. non-IID partition, $\eta = 0.05$ on CIFAR10. (**a**) Test accuracy. $C = 0.1$. (**b**) Test accuracy. $C = 0.5$. (**c**) Training loss. $C = 0.1$. (**d**) Training loss. $C = 0.5$.

In conclusion, by taking the advantage of previous unexploited client gradients, the rate of convergence can be greatly expedited with FEDUMF compared to the vanilla FEDAVG.

6. Conclusions

In this paper, we have proposed a federated learning algorithm—FEDUMF—that allows unselected clients to train model update and fuses the resulting new model later when the clients receive authorization to upload. This design can avoid wasting computation capacity and expedite the convergence rate of FL. A rigorous convergence analysis was given for FEDUMF, which proved a faster convergence rate than vanilla FEDAVG. The simulation performance demonstrated that FEDUMF performs better than the traditional FEDAVG, where unselected clients are idle. It also shows satisfactory results in combination with the SOTA algorithms.

Author Contributions: Software, B.L.; Writing—original draft, B.L.; Writing—review & editing, B.L., S.C. and K.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (KAKENHI) under Grant JP18K18044, the Ji'An Finance & Science Foundation under Grant No. [2019]55, and the Grant for Ji'An Key Lab for computer-aided diagnosis of mental disease.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Additional Experiments

To better illustrate the generality of our algorithm, as well as its superiority, we conducted a comparative experiment combining FedUmf with the state-of-the-art algorithm (FedDC), compared with existing algorithms as baselines, for example, FedAvg, FedProx [19], Scaffold [35], and FedDyn [36], FedDC [37].

We used the same experimental setup as some state-of-the-art algorithms and conducted rigorous and extensive experimental demonstrations on five open-source datasets, including MNIST [38], EMNIST-L(EMNIST) [39], CIFAR10, CIFAR100 [40], and synthetic datasets [19]. For the experimental setting, we adopted the same training method, dataset, and classification. In the IID case, the training set was randomly and equally distributed among all specified clients. In the unbalanced distribution, we used a different lognormal distribution sampling with a standard deviation of 0.3 to ensure that the total amount was not equal under the random distribution of the sample. Under the non-IID case, we adopted the Dirichlet method [41] to generate the label heterogeneous distribution, which we set as non-IID. The representative parameters were set to 0.6. For synthetic datasets, we still used the previous three different data allocation settings; we use Synthetic (0, 0), Synthetic (0, 1), and Synthetic (1, 0) Three heterogeneous assignments were used as the experimental setup.

We set the clients' local training batch size to 50. The authorized number of training rounds was 1 round, the learning rate was 0.1, and the learning rate decay was 0.998 per round. We used the same setup as in the previous work in order to cross-check the literature to compare the results. In the baseline algorithm settings, the parameters used by FedProx are set to $\mu = 1e-4$, the parameters of FedDyn are set to $\alpha = 1e-2$, and the parameters of FedDC are set to $\alpha = 1e-2$. For the synthetic dataset, we follow the [37] settings, setting the total number of clients as 20 and the local training batch as 10.

As pointed out in the literature [37], we used a simple fully-connected network (FCN) on MNIST and EMNIST. The slight difference is that although both contain only two fully connected layers, MNIST uses 100 neurons per layer, while EMNIST uses 200 neurons. We used a convolutional neural network (CNN) on CIFAR10 and CIFAR100. CNN consists of

two convolutional layers with a kernel size of 5∗5. It is followed by two fully connected layers, containing 384 and 192 parameters, respectively.

We observe from Tables A1 and A2 that our algorithm has strong gains as well as generality; more detailed training curves are given in Figures A1–A5.

Table A1. The test accuracy (%) on IID, non-IID, and unbalanced data for partial client participation (15% and 65%). "**ACCURACY**" is recorded as first accuracy.

Methods	**FedAvg**	**FedProx**	**SCAFFOLD**	**FedDyn**	**FedDC**	**FedUmf**
Setting	100 clients 15% participation					
MNIST-iid	96.90	96.90	96.99	97.779	98.13	**98.25**
MNIST-unbalance	97.06	97.01	96.87	97.66	98.09	**98.20**
MNIST-noniid	96.45	96.44	96.85	97.56	98.09	**98.21**
EMNIST-iid	92.22	92.12	92.48	93.51	94.30	**94.98**
EMNIST-unbalance	92.41	92.31	92.01	93.48	94.15	**94.78**
EMNIST-noniid	91.36	91.26	92.06	93.36	94.13	**94.16**
CIFAR10-iid	74.12	74.24	74.1	81.90	81.89	**82.90**
CIFAR10-unbalance	74.14	74.25	72.30	81.77	81.92	**82.45**
CIFAR10-noniid	73.72	73.79	73.55	78.83	78.17	**81.35**
CIFAR100-iid	36.6	36.54	36.04	50.68	52.05	**53.29**
CIFAR100-unbalance	38.7	38.67	34.76	51.07	52.01	**53.58**
CIFAR100-noniid	35.92	36.15	35.88	43.08	43.43	**51.11**
Setting	100 clients 65% participation					
MNIST-iid	96.98	96.96	97.06	98.01	98.34	**98.40**
MNIST-unbalance	97.20	97.20	96.88	97.90	98.33	**98.34**
MNIST-noniid	96.56	96.56	96.88	97.90	98.30	**98.33**
EMNIST-iid	92.35	92.46	92.45	93.75	94.86	**95.25**
EMNIST-unbalance	92.53	92.58	92.25	93.75	94.71	**94.98**
EMNIST-noniid	91.65	91.71	92.27	93.83	94.65	**95.13**
CIFAR10-iid	79.86	80.20	83.13	82.62	83.64	**84.39**
CIFAR10-unbalance	80.21	80.30	82.81	82.90	83.40	**84.54**
CIFAR10-noniid	78.34	78.41	81.73	81.92	82.42	**82.89**
CIFAR100-iid	36.69	36.42	47.30	47.10	52.53	**54.48**
CIFAR100-unbalance	38.12	38.10	48.94	47.22	52.92	**54.23**
CIFAR100-noniid	37.79	38.02	46.75	47.42	51.42	**54.43**

Table A2. Different strategies are used to achieve the specified accuracy. Includes one IID setting, one unbalanced setting, and the non-IID setting. The non-IID Dirichlet coefficient is set to 0.6. com R□ indicates the communication rounds that reach the target. Speed ⇑ indicates acceleration relative to baseline.

Model	100 Clients 15% Participation						100 Clients 65% Participation					
	iid		unbalance		noniid		iid		unbalance		noniid	
	com R□	Speed ⇑	com R□	Speed⇑	com R□	Speed ⇑	com R□	Speed ⇑	com R□	Speed⇑	com R□	Speed ⇑
MNIST, Target accuracy 95%												
FedAvg	40	-	34	-	52	-	36	-	32	-	59	-
FedProx	40	1.00×	35	0.97×	52	1.00×	36	1.00×	31	1.03×	49	1.20×
SCAFFOLD	39	1.03×	42	0.81×	43	1.21×	35	1.03×	38	0.84×	39	1.51×
FedDyn	25	1.60×	25	1.36×	31	1.68×	11	3.27×	16	2.00×	16	3.69×
FedDC	22	1.82×	28	1.21×	24	2.17×	25	1.44×	14	2.29×	14	4.21×
FedUmf	11	3.64×	12	2.83×	14	3.71×	10	3.60×	10	3.20×	12	4.92×
EMNIST, Target accuracy 91%												
FedAvg	67	-	68	-	87	-	62	-	57	-	83	-
FedProx	69	0.97×	68	1.00×	92	0.95×	62	1.00×	58	0.98×	81	1.02×
SCAFFOLD	64	1.05×	72	0.94×	74	1.18×	62	1.00×	68	0.84×	67	1.24×
FedDyn	39	1.72×	39	1.74×	55	1.58×	29	2.14×	28	2.04×	36	2.31×
FedDC	40	1.68×	34	2.00×	46	1.89×	23	2.70×	23	2.48×	28	2.96×
FedUmf	21	3.19×	24	2.83×	33	2.64×	19	3.26×	19	3.00×	21	3.95×
CIFAR10, Target accuracy 70%												
FedAvg	197	-	180	-	225	-	50	-	50	-	74	-
FedProx	201	0.98×	179	1.01×	225	1.00×	50	1.00×	50	1.00×	74	1.00×
SCAFFOLD	213	0.92×	236	0.76×	225	1.0×	37	1.35×	51	0.98×	69	1.07×
FedDyn	70	2.81×	71	2.54×	125	1.8×	42	1.19×	50	1.00×	69	1.07×
FedDC	69	2.86×	72	2.50×	125	1.80×	31	1.61×	50	1.00×	50	1.48×
FedUmf	50	3.94×	51	3.53×	62	3.63×	28	1.79×	40	1.25×	46	1.61×
CIFAR100, Target accuracy 35%												
FedAvg	198	-	165	-	203	-	213	-	149	-	175	-
FedProx	198	1.00×	165	1.00×	203	1.00×	213	1.0×	148	1.01×	176	0.99×
SCAFFOLD	207	0.96×	220	0.75×	205	0.99×	69	3.09×	70	2.13×	73	2.40×
FedDyn	75	2.64×	76	2.17×	109	1.86×	100	2.13×	99	1.51×	103	1.70×
FedDC	74	2.68×	72	2.29×	105	1.93×	77	2.77×	73	2.04×	78	2.24×
FedUmf	63	3.14×	65	2.54×	70	2.90×	40	5.33×	49	3.04×	50	3.50×

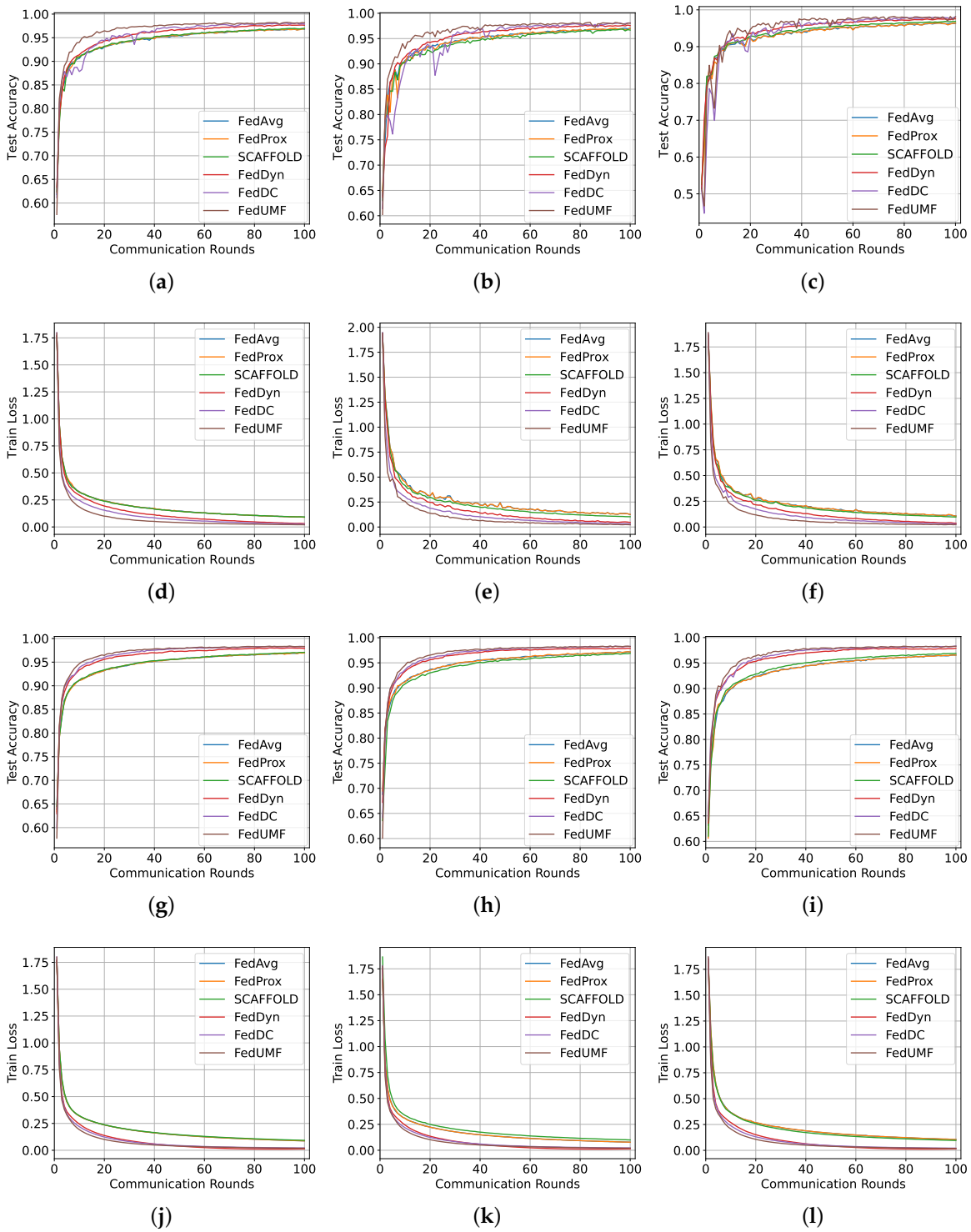

Figure A1. Convergence and loss plots of FedUMF for different setting with 100 clients adopting 15% and 65% client participating settings on MNIST. (**a**) Test Acc. IID 15%. (**b**) Test Acc. unbalance 15%. (**c**) Test Acc. Drichlet 0.6 15%. (**d**) Train Loss. IID 15%. (**e**) Train Loss. Drichlet 0.3 15%. (**f**) Train Loss. Drichlet 0.6 15%. (**g**) Test Acc. IID 65%. (**h**) Test Acc. unbalance 65%. (**i**) Test Acc. Drichlet 0.6 65%. (**j**) Train Loss. IID 65%. (**k**) Train Loss. unbalance 65%. (**l**) Train Loss. Drichlet 0.6 65%.

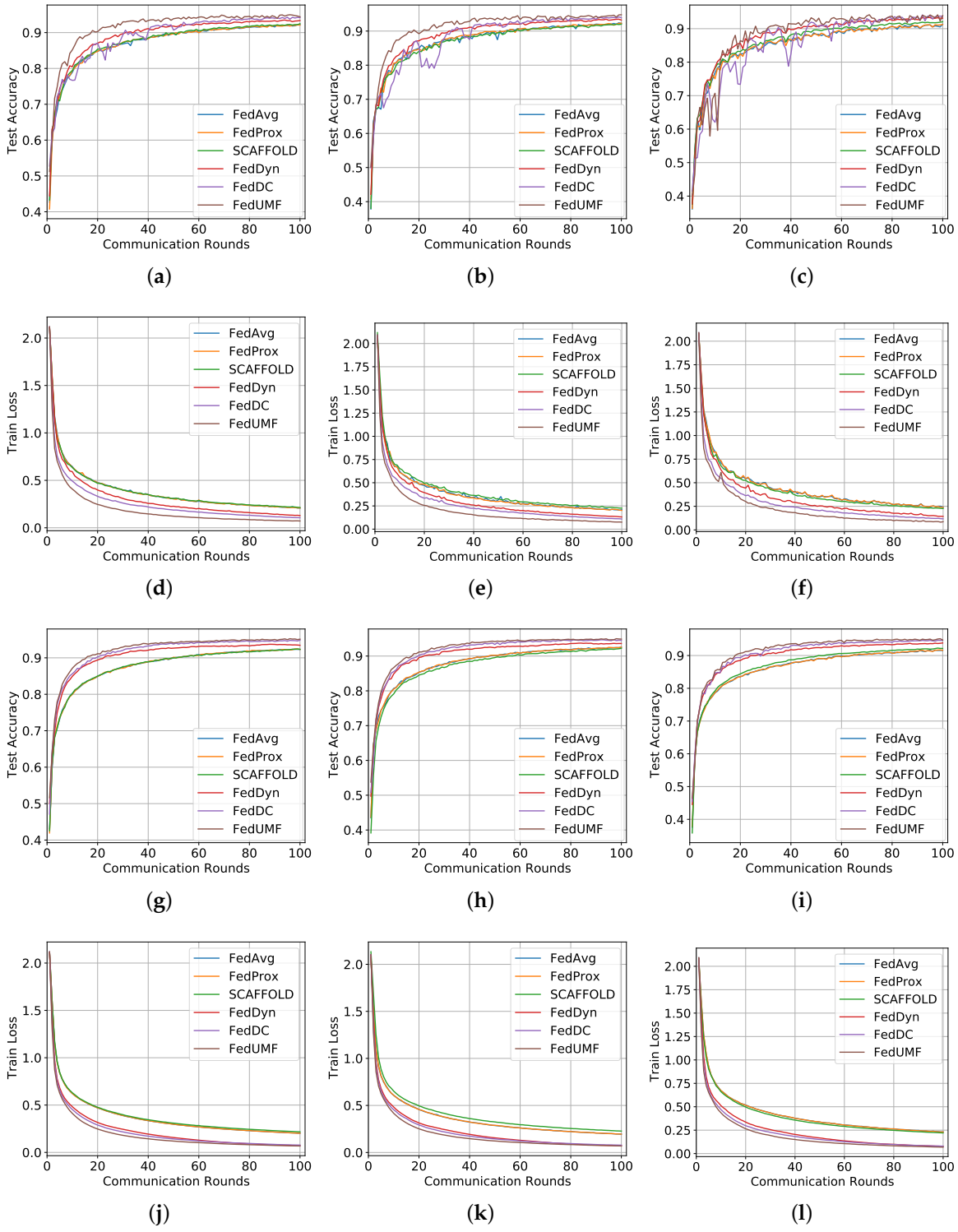

Figure A2. Convergence and loss plots of FedUMF for different setting with 100 clients adopting 15% and 65% client participating settings on EMNIST. (**a**) Test Acc. IID 15%. (**b**) Test Acc. unbalance 15%. (**c**) Test Acc. Drichlet 0.6 15%. (**d**) Train Loss. IID 15%. (**e**) Train Loss. unbalance 15%. (**f**) Train Loss. Drichlet 0.6 15%. (**g**) Test Acc. IID 65%. (**h**) Test Acc. unbalance 65%. (**i**) Test Acc. Drichlet 0.6 65%. (**j**) Train Loss. IID 65%. (**k**) Train Loss. unbalance 65%. (**l**) Train Loss. Drichlet 0.6 65%.

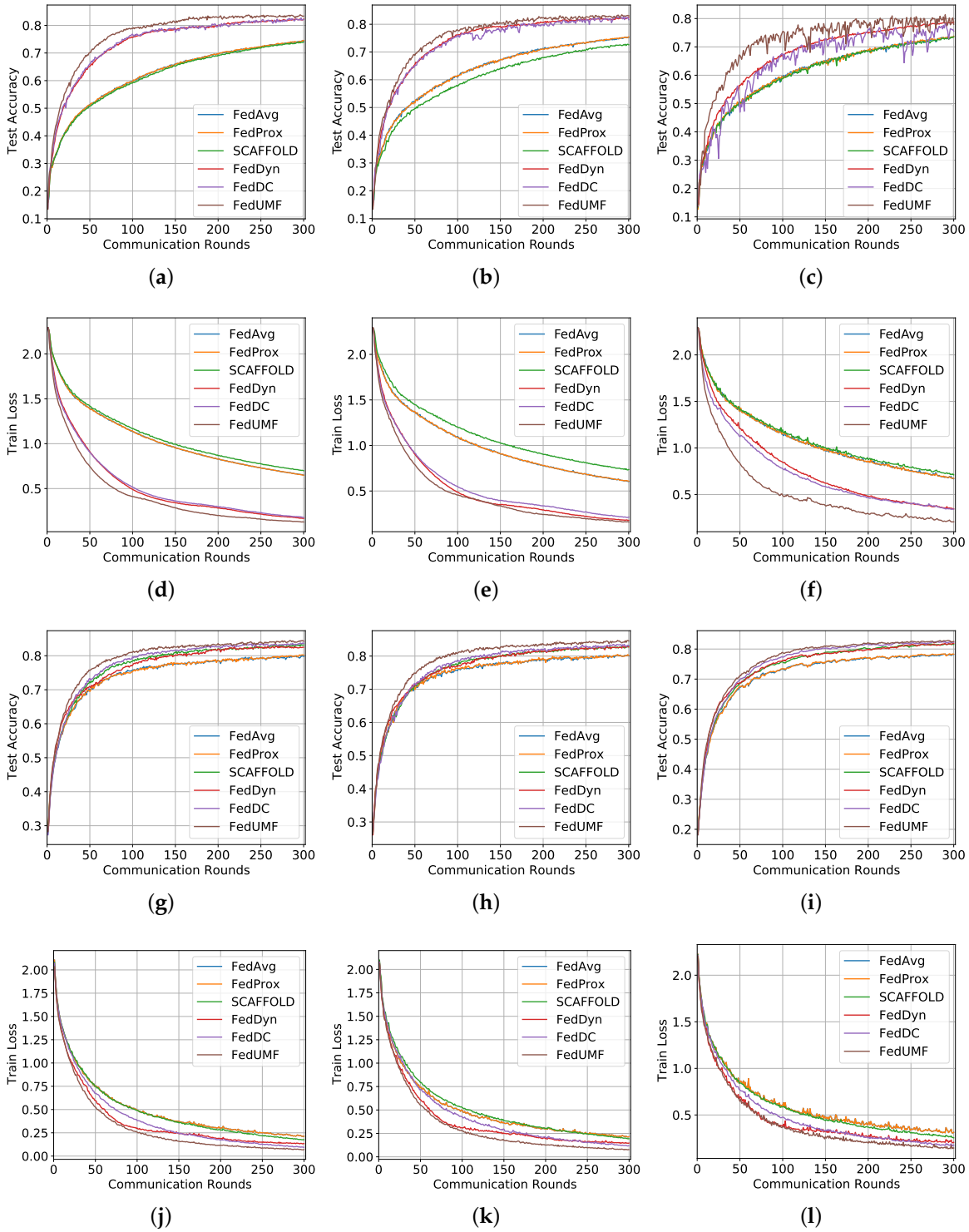

Figure A3. Convergence and loss plots of FedUMF for different setting with 100 clients adopting 15% and 65% client participating settings on CIFAR. (**a**) Test Acc. IID 15%. (**b**) Test Acc. unbalance 15%. (**c**) Test Acc. Drichlet 0.6 15%. (**d**) Train Loss. IID 15%. (**e**) Train Loss. unbalance 15%. (**f**) Train Loss. Drichlet 0.6 15%. (**g**) Test Acc. IID 65%. (**h**) Test Acc. unbalance 65%. (**i**) Test Acc. Drichlet 0.6 65%. (**j**) Train Loss. IID 65%. (**k**) Train Loss. unbalance 65%. (**l**) Train Loss. Drichlet 0.6 65%.

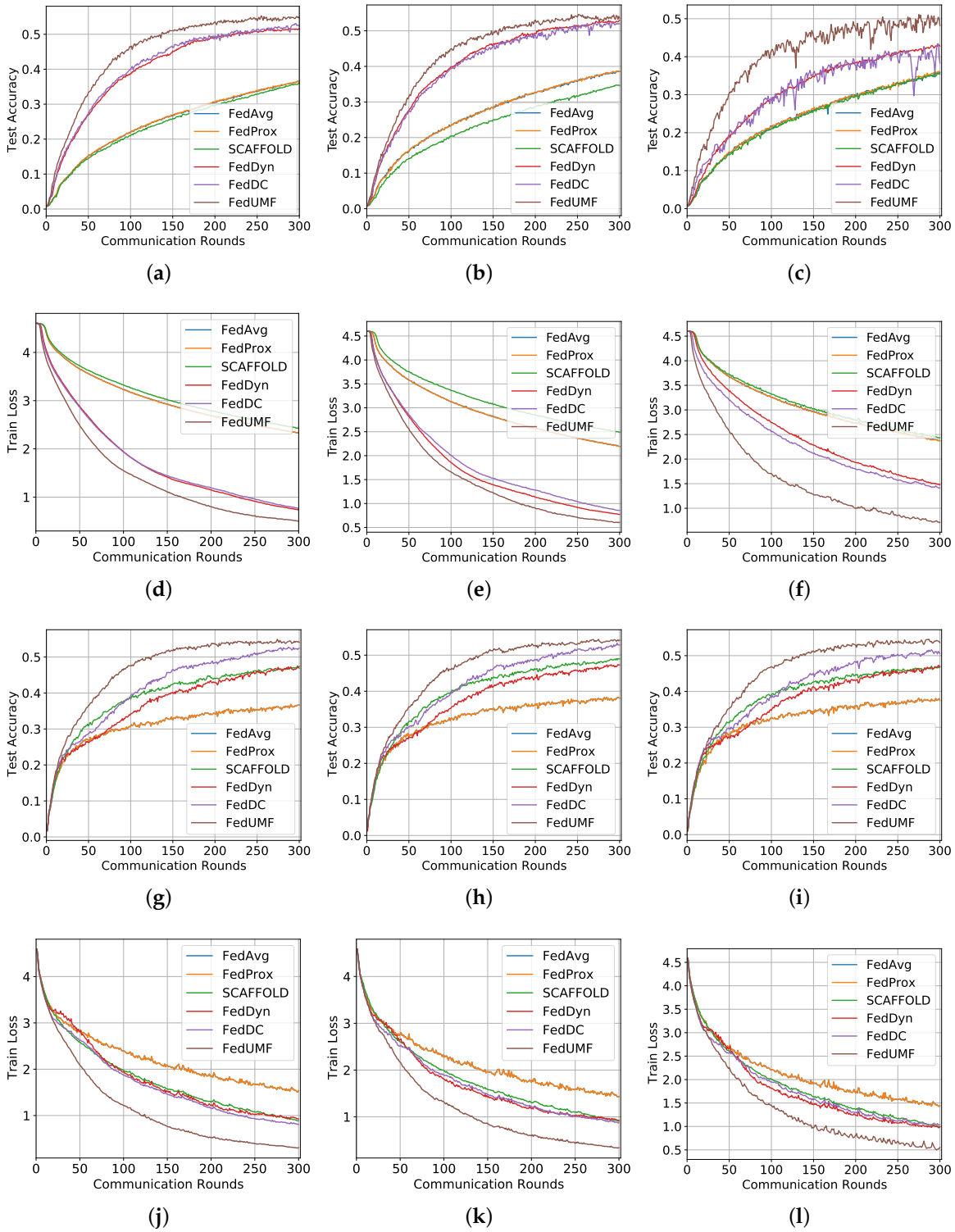

Figure A4. Convergence and loss plots of FedUMF for different setting with 100 clients adopting 15% and 65% client participating settings on CIFAR100. (**a**) Test Acc. IID 15%. (**b**) Test Acc. unbalance 15%. (**c**) Test Acc. Drichlet 0.6 15%. (**d**) Train Loss. IID 15%. (**e**) Train Loss. unbalance 15%. (**f**) Train Loss. Drichlet 0.6 15%. (**g**) Test Acc. IID 65%. (**h**) Test Acc. unbalance 65%. (**i**) Test Acc. Drichlet 0.6 65%. (**j**) Train Loss. IID 65%. (**k**) Train Loss. unbalance 65%. (**l**) Train Loss. Drichlet 0.6 65%.

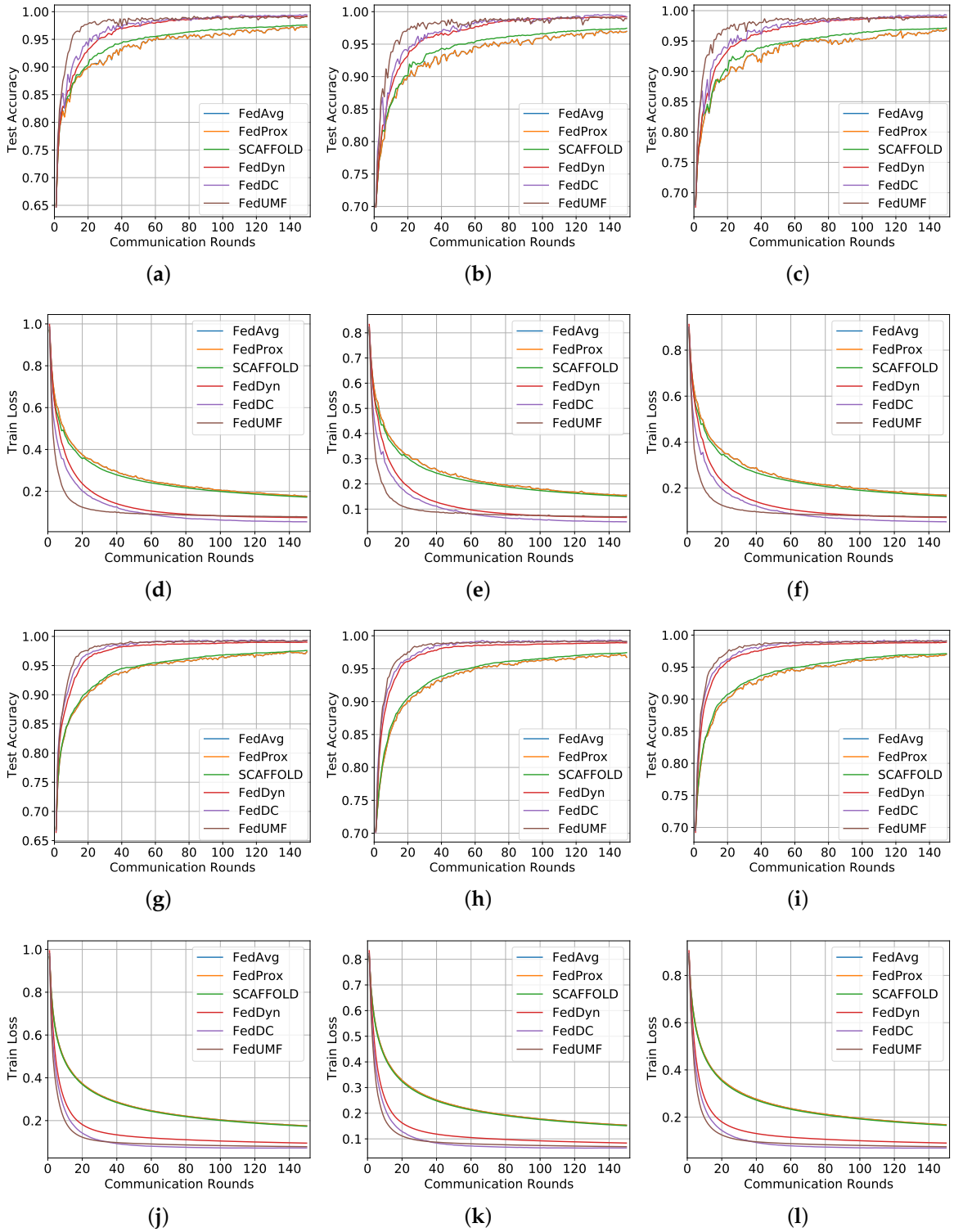

Figure A5. Convergence and loss plots on the synthetic dataset. There are three types of settings: the homogeneous setting with (0, 0), and two heterogeneous settings with (1, 0) and (0, 1), respectively. (**a**) Synthetic (0, 0) 15% Test Acc. (**b**) Synthetic (0, 1) 15% Test Acc. (**c**) Synthetic (1, 0) 15% Test Acc. (**d**) Synthetic (0, 0) 15% Train Loss. (**e**) Synthetic (0, 1) 15% Train Loss. (**f**) Synthetic (1, 0) 15% Train Loss. (**g**) Synthetic (0, 0) 65% Test Acc. (**h**) Synthetic (0, 1) 65% Test Acc. (**i**) Synthetic (1, 0) 65% Test Acc. (**j**) Synthetic (0, 0) 65% Train Loss. (**k**) Synthetic (0, 1) 65% Train Loss. (**l**) Synthetic (1, 0) 65% Train Loss.

References

1. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konečný, J.; Mazzocchi, S.; McMahan, B.; et al. Towards federated learning at scale: System design. In Proceedings of the 2nd SysML Conference, Stanford, CA, USA, 31 March–2 April 2019; pp. 1–15.
2. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
3. Konecný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. In Proceedings of the NIPS Workshop on Private Multi-Party Machine Learning, Barcelona, Spain, 9 December 2016.
4. Robbins, H.; Monro, S. A Stochastic Approximation Method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [CrossRef]
5. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Zhao, S. Advances and open problems in federated learning. *arXiv* **2019**, arXiv:1912.04977.
6. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
7. Imteaj, A.; Thakker, U.; Wang, S.; Li, J.; Amini, M.H. A Survey on Federated Learning for Resource-Constrained IoT Devices. *IEEE Internet Things J.* **2022**, *9*, 1–24. [CrossRef]
8. Bernstein, J.; Wang, Y.; Azizzadenesheli, K.; Anandkumar, A. signSGD: Compressed optimisation for non-convex problems. *arXiv* **2018**, arXiv:1802.04434.
9. Wen, W.; Xu, C.; Yan, F.; Wu, C.; Wang, Y.; Chen, Y.; Li, H. TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1509–1519.
10. Woodworth, B.E.; Wang, J.; Smith, A.; McMahan, B.; Srebro, N. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; pp. 8496–8506.
11. Zhou, F.; Cong, G. On the Convergence Properties of a *K*-step Averaging Stochastic Gradient Descent Algorithm for Nonconvex Optimization. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 3219–3227.
12. Sattler, F.; Wiedemann, S.; Müller, K.; Samek, W. Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [CrossRef]
13. Karimireddy, S.P.; Rebjock, Q.; Stich, S.U.; Jaggi, M. Error Feedback Fixes SignSGD and other Gradient Compression Schemes. *arXiv* **2019**, arXiv:1901.09847.
14. Reisizadeh, A.; Mokhtari, A.; Hassani, H.; Pedarsani, R. An Exact Quantized Decentralized Gradient Descent Algorithm. *IEEE Trans. Signal Process.* **2019**, *67*, 4934–4947. [CrossRef]
15. Yuan, J.; Xu, M.; Ma, X.; Zhou, A.; Liu, X.; Wang, S. Hierarchical Federated Learning through LAN-WAN Orchestration. *arXiv* **2020**, arXiv:2010.11612.
16. Dai, X.; Yan, X.; Zhou, K.; Yang, H.; Ng, K.K.W.; Cheng, J.; Ling Fan, Y. Hyper-Sphere Quantization: Communication-Efficient SGD for Federated Learning. *arXiv* **2019**, arXiv:1911.04655.
17. Konecný, J. Stochastic, Distributed and Federated Optimization for Machine Learning. *arXiv* **2017**, arXiv:1707.01155.
18. Yu, Y.; Wu, J.; Huang, L. Double Quantization for Communication-Efficient Distributed Optimization. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 4440–4451.
19. Sahu, A.K.; Li, T.; Sanjabi, M.; Zaheer, M.; Talwalkar, A.; Smith, V. On the Convergence of Federated Optimization in Heterogeneous Networks. *arXiv* **2018**, arXiv:1812.06127.
20. Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; Vojnovic, M. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1709–1720.
21. Jiang, P.; Agrawal, G. A Linear Speedup Analysis of Distributed Deep Learning with Sparse and Quantized Communication. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; pp. 2525–2536.
22. Lu, X.; Liao, Y.; Liu, C.; Lio, P.; Hui, P. Heterogeneous Model Fusion Federated Learning Mechanism Based on Model Mapping. *IEEE Internet Things J.* **2022**, *9*, 6058–6068. [CrossRef]
23. Yu, L.; Albelaihi, R.; Sun, X.; Ansari, N.; Devetsikiotis, M. Jointly Optimizing Client Selection and Resource Management in Wireless Federated Learning for Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 4385–4395. [CrossRef]
24. Wu, W.; He, L.; Lin, W.; Mao, R.; Maple, C.; Jarvis, S.A. SAFA: A Semi-Asynchronous Protocol for Fast Federated Learning with Low Overhead. *IEEE Trans. Comput.* **2020**, *70*, 655–668. [CrossRef]
25. Zhang, T.; Song, A.; Dong, X.; Shen, Y.; Ma, J. Privacy-Preserving Asynchronous Grouped Federated Learning for IoT. *IEEE Internet Things J.* **2022**, *9*, 5511–5523. doi: 10.1109/JIOT.2021.3111088. [CrossRef]
26. Zhang, W.; Gupta, S.; Lian, X.; Liu, J. Staleness-aware Async-SGD for Distributed Deep Learning. *arXiv* **2015**, arXiv:1511.05950.
27. Xie, C.; Koyejo, S.; Gupta, I. Asynchronous Federated Optimization. *arXiv* **2019**, arXiv:1903.03934.

28. Chen, Y.; Nin, Y.; Slawski, M.; Rangwala, H. Asynchronous Online Federated Learning for Edge Devices. *arXiv* **2019**. arXiv:1911.02134.
29. Zheng, S.; Meng, Q.; Wang, T.; Chen, W.; Yu, N.; Ma, Z.M.; Liu, T.Y. Asynchronous Stochastic Gradient Descent with Delay Compensation. In *Proceedings of the Machine Learning Research*; PMLR International Convention Centre: Sydney, Australia, 2017; Volume 70, pp. 4120–4129.
30. Reisizadeh, A.; Mokhtari, A.; Hassani, H.; Jadbabaie, A.; Pedarsani, R. FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. In Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), Palermo, Italy, 26–28 August 2020.
31. Stich, S.U. Local SGD Converges Fast and Communicates Little. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 March–3 May 2018.
32. Wang, J.; Joshi, G. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. *arXiv* **2018**, arXiv:1808.07576.
33. Yu, H.; Yang, S.; Zhu, S. Parallel restarted SGD for non-convex optimization with faster convergence and less communication. *arXiv* **2018**, arXiv:1807.06629.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the CVPR, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
35. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.J.; Stich, S.U.; Suresh, A.T. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. *arXiv* **2019**, arXiv:1910.06378.
36. Acar, D.A.E.; Zhao, Y.; Matas, R.; Mattina, M.; Whatmough, P.; Saligrama, V. Federated Learning Based on Dynamic Regularization. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 3–7 May 2021.
37. Gao, L.; Fu, H.; Li, L.; Chen, Y.; Xu, M.; Xu, C.Z. FedDC: Federated Learning with Non-IID Data via Local Drift Decoupling and Correction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022.
38. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
39. Cohen, G.; Afshar, S.; Tapson, J.; van Schaik, A. EMNIST: Extending MNIST to handwritten letters. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 2921–2926. [CrossRef]
40. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, University of Tronto, Toronto, ON, Canada, 2009.
41. Lin, T.; Karimireddy, S.P.; Stich, S.U.; Jaggi, M. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. *arXiv* **2021**, arXiv:2102.04761.