electronics

MDPI

Article

Clustered Federated Learning Based on Momentum Gradient Descent for Heterogeneous Data

Xiaoyi Zhao, Ping Xie *, Ling Xing, Gaoyuan Zhang and Huahong Ma

School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China; 200320050346@stu.haust.edu.cn (X.Z.); xingling_my@163.com (L.X.); zhanggaoyuan407@163.com (G.Z.); mhh@haust.edu.cn (H.M.)

* Correspondence: xieping@haust.edu.cn

Abstract: Data heterogeneity may significantly deteriorate the performance of federated learning since the client's data distribution is divergent. To mitigate this issue, an effective method is to partition these clients into suitable clusters. However, existing clustered federated learning is only based on the gradient descent method, which leads to poor convergence performance. To accelerate the convergence rate, this paper proposes clustered federated learning based on momentum gradient descent (CFL-MGD) by integrating momentum and cluster techniques. In CFL-MGD, scattered clients are partitioned into the same cluster when they have the same learning tasks. Meanwhile, each client in the same cluster utilizes their own private data to update local model parameters through the momentum gradient descent. Moreover, we present gradient averaging and model averaging for global aggregation, respectively. To understand the proposed algorithm, we also prove that CFL-MGD converges at an exponential rate for smooth and strongly convex loss functions. Finally, we validate the effectiveness of CFL-MGD on CIFAR-10 and MNIST datasets.

Keywords: clusters; data heterogeneity; federated learning; momentum gradient descent (MGD)

Citation: Zhao, X.; Xie, P.; Xing, L.; Zhang, G.; Ma, H. Clustered Federated Learning Based on Momentum Gradient Descent for Heterogeneous Data. *Electronics* **2023**, *12*, 1972. https://doi.org/10.3390/electronics12091972

Academic Editors: Juan M. Corchado, Byung-Gyu Kim, Carlos A. Iglesias, In Lee, Fuji Ren and Rashid Mehmood

Received: 22 March 2023
Revised: 20 April 2023
Accepted: 21 April 2023
Published: 24 April 2023

1. Introduction

Recently, machine learning [1,2] has been successfully applied in distinct fields, such as computer vision [3], voice recognition [4], and natural language processing [5]. Large amounts of data are required in these data-intensive applications. Nonetheless, data are generally generated and stored on personal terminal devices, such as mobile phones, personal computers, wearable devices, etc. Traditional machine learning collects data in a centralized manner and stores the data in a data center. However, this approach no longer meets the requirements for privacy. Protecting privacy [6] while collecting large amounts of data remains a key issue. For this reason, federated learning (FL) [7–9] was proposed by Google in 2016. FL, as a promising edge-learning framework [10], has received widespread attention as a distributed paradigm. FL enables multiple clients to collaboratively train a global model without sharing or exchanging their own private data. During the process of model training, information related to the model or information in encrypted form can be exchanged between parties. This exchange does not expose any protected private parts of the data on each client, and efficient learning can be carried out between multiple participants and compute nodes. The emergence of federated learning can effectively balance the contradiction between benefit and privacy, solving the problem of data aggregation [11]. However, because of the highly decentralized system architecture of FL, it also encounters a crucial challenge—data heterogeneity [12,13].

In FL, data heterogeneity mainly stems from the fact that each client participating in the training is independently distributed, but does not follow the same sampling method, resulting in non-independent identically distributed (non-i.i.d.) data [14,15]. This problem leads to a steep decline in model accuracy, and how to mitigate the adverse effects of non-i.i.d. is an open question. References [16,17] proposed the *k*-means clustered algorithm

based on FL. However, the *k*-means clustering algorithm can only be used on convex datasets, meaning that the shape of the *k*-means cluster can only be spherical, which cannot be generalized to arbitrary shapes. Additionally, it heavily depends on the value of *k*, and when the data amounts are large, we cannot judge in advance. Reference [16] used the cosine distance to measure the similarities between network data objects. Reference [18] proposed StoCFL, a novel clustered federated learning approach for addressing generic non-IID issues. They also employed cosine distance to measure the similarity of clients. However, their method did not account for differences in data values, in practice. Reference [17] proposed one-shot federated clustering by using *k*-means to cluster the clients [19]. Nevertheless, with the *k*-means method, is difficult to determine the value of *k* in practical applications. In addition, some recent works have utilized user-clustered methods to exploit data heterogeneity.

Reference [20] proposed a user-clustered algorithm based on the similarity between clients, Reference [21] investigated a user-clustered algorithm based on model parameter distance, and Reference [22] developed a user-clustered algorithm based on the weighted sum of clients. The aforementioned works based on the user-clustered algorithm merely considered the conditions to achieve clustering; however, the convergence rate has rarely been a concern. Additionally, Reference [23] proposed the IFCA algorithm to divide clusters by minimizing loss functions and provided convergence analysis, but the convergence rate is not desirable. Reference [24] introduced a new clustered FL algorithm based on weighted clients and discussed the convergence analysis. However, in cases with multiple missing attributes, different weights must be assigned to the missing combinations of different attributes, which significantly increases the calculation difficulty and reduces the prediction accuracy. Despite the numerous works that have utilized clustered methods to address data heterogeneity in FL, the convergence rate of clustered algorithms in FL remains an urgent problem.

Currently, as an acceleration technique, the momentum method is widely used to improve the convergence rate of the optimization algorithm [25–27]. Common algorithms include momentum gradient descent (MGD) [28] and adaptive algorithms, such as AdaGrad [29], Adam [30–32], as well as some subsequently improved algorithms [33,34]. By incorporating momentum, the previous gradient is reused, and a cumulative discount is applied to it. By modifying the direction of gradient improvement, the influence of the previous gradient on the current gradient is used to accelerate the training speed of the model. The momentum method used in model training can effectively reduce the training time and accelerate the convergence of the algorithm. For instance, Polyak's momentum GD is a classic momentum GD used to train neural networks, which has good generalization and fast convergence [35–37]. As shown in Figure 1 [25], the MGD algorithm can accelerate gradient descent and alleviate oscillation amplitude. In recent years, the momentum method has been widely used in optimization algorithms to compensate for the slow convergence in FL. Nevertheless, research on the application of momentum methods in federated learning based on clustered algorithms is lacking.

In order to accelerate the convergence of the algorithm and alleviate data heterogeneity, we propose a framework named clustered federated learning based on momentum gradient descent (CFL-MGD), as shown in Figure 2. It combines Polyak's momentum [38] and clustered methods for use in federated learning. The learning process involves selecting locally optimal parameters by minimizing the local loss function and dividing clients with the same learning parameters into a cluster. The momentum gradient descent algorithm is then used to update the model during the training process.

The essential contributions of this paper are as follows:

- We propose clustered federated learning based on momentum gradient descent to address the data owned by clients in federated learning, where the data are habitually non-independently identically distributed.
- We establish a convergence guarantee for the proposed CFL-MGD algorithm in a convex setting and show that our algorithm can achieve exponential convergence.

- We experimentally show that the proposed algorithm can perform well in non-convex settings, such as neural networks.

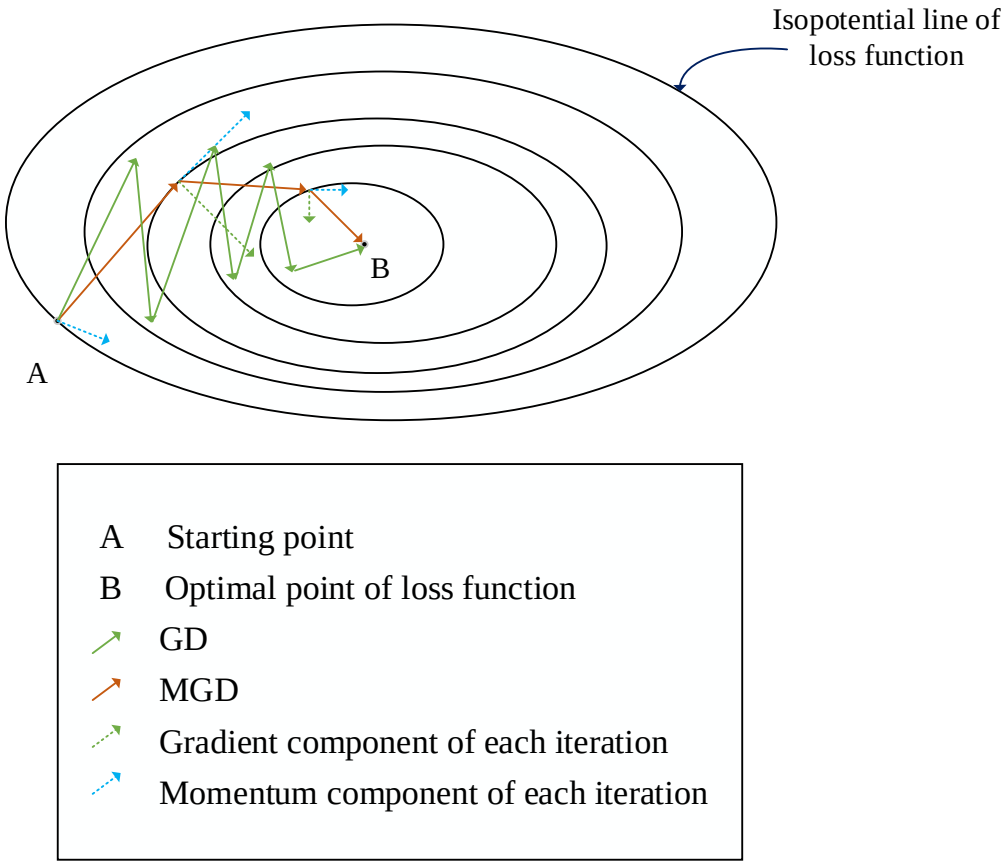


Figure 1. Comparison of MGD and GD.

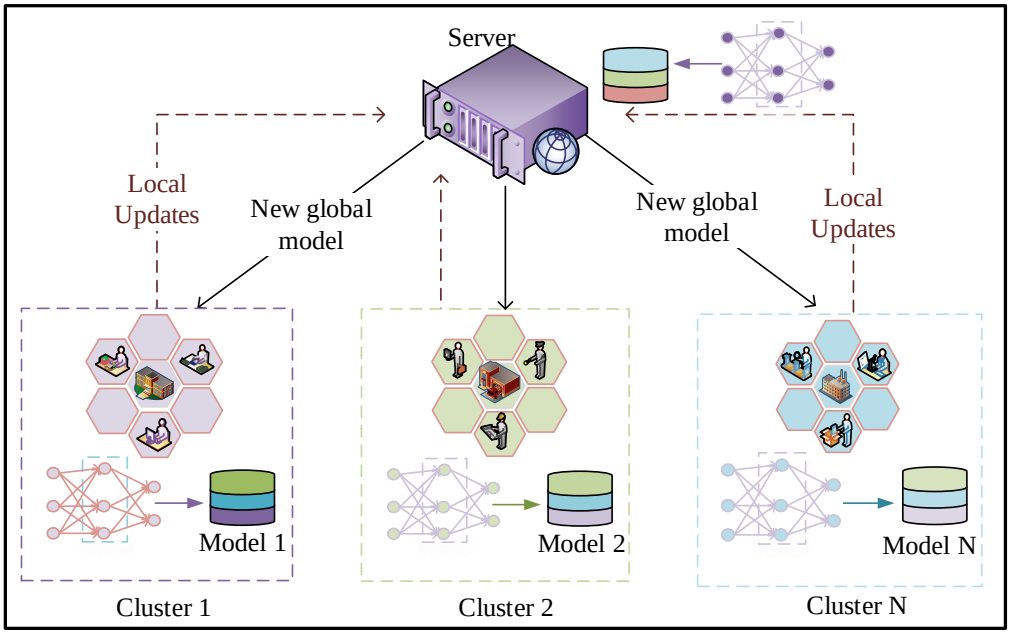


Figure 2. The overall architecture of clustered federated learning systems.

The abbreviation interpretation is shown in Table 1.

Table 1. Abbreviation summary table.

Abbreviation	Definition
m	the number of clients
P	total number of clusters
C_p	the p-th of the cluster
M_t	randomly select participating clients
Y	a subset of the data points on the i-th client
$\|Y\|$	the number of data points on the i-th client
$D_i, \|D_i\|$	the dataset and dataset size of client i
$\phi_i(\chi; Y)$	the i-th client's loss function
$\phi_p(\chi)$	the p-th cluster loss function
$[P]$	the set of integers $\{1, 2, \ldots, P\}$
$\|\cdot\|$	the ℓ_2 norm of vectors
$r_{i,p}$	$r_{i,p} = 1$ if $i \in C_p$ else $r_{i,p} = 0$
g_i	unbiased gradient estimation of $\nabla\phi_i(w)$
η, β	learning rate
T	the number of epochs

The remainder of this paper is established as follows. In Section 2, we introduce the related work and define the preliminaries in Section 3. Then, we propose the algorithm in Section 4 and analyze theoretical guarantees in Section 5. Next, in Section 6, we carefully verify our theoretical analysis through experiments. Ultimately, we conclude this paper in Section 7.

2. Related Work

Federated learning faces the challenge of data heterogeneity due to its highly decentralized architecture. Numerous studies have been conducted to mitigate this issue. Clustered FL was proposed in [20] as a solution for processing data heterogeneity. Subsequently, many studies have shown that clustered federated learning is an effective way to address the problem of data heterogeneity [14,39,40]. It divides clients into two partitions based on the cosine similarity between clients. Reference [23] proposed the IFCA algorithm to estimate cluster classes by minimizing the loss functions. In our paper, our algorithm is similar to IFCA, but we delivered a major contribution to the convergence rate of the algorithm. Meanwhile, the convergence of our algorithm and generalization of test data are guaranteed.

2.1. FL and Data Heterogeneity

Federated learning consists of a central server and multiple clients. In FL, users' personal devices are often in various regions and industries, so the data owned by each client are non-i.i.d., resulting in data heterogeneity. In order to alleviate the problem, Reference [41] proposed a combinatorial approach involving personalized federated learning. However, when extended to extensive networks, this approach is limited to convex targets. User clustering methods are proposed in [42], but the parallel algorithm of user clustering based on assumption loss is not common, and convergence analysis has not been discussed. Reference [21] proposed a user-clustered algorithm based on the model parameter distance, FeSEM, but the convergence rate of the algorithm was not analyzed. Reference [17] proposed one-shot federated clustering by using the clustering method (using local k-means), but it only performed one round of communication, resulting in poor accuracy. Clustered federated learning (CFL) was proposed in [20]; it divided clients into two partitions based on the similarity between clients, but it paid little attention to the convergence rate. Reference [43] proposed proportional fairness-clustered federated learning and provided a detailed convergence analysis, but the convergence effect was closely related to the assumption that global variance is bounded.

2.2. Momentum Method

As an acceleration technique, the momentum method is widely used to improve the convergence rate of optimization algorithms. It is a special first-order optimization method based on the classical gradient descent [25] method by adding momentum. Momentum algorithms are divided into two classes: the heavy-ball momentum method proposed by Polyak in 1964 [38], and the Nesterov-accelerated gradient (NAG) proposed by Nesterov in 1983 [44]. Ghadimi et al. conducted in-depth studies on the convergence of the heavy-ball method and gave the average and individual convergence rates under the condition of a smooth objective function [45]; however, they did not reach the optimal convergence rate. Reference [46] established an algorithm framework with multiple parameters, which uniformly processed the gradient descent method, the heavy-ball method, and the NAG method. In this framework, different optimization algorithms could be obtained by setting different parameters. The momentum method has been widely used to improve the convergence rate of optimization algorithms. Therefore, it is feasible to apply momentum in federated learning based on clustered algorithms.

3. Preliminaries

Let us consider a clustered federated learning framework with P clusters and m clients. We have

$$\min_{\chi_p} \phi_p(\chi_p) = \frac{|D_p|}{|D|} \sum_{i \in C_p^*} \phi_i(\chi_p), \tag{1}$$

where $p \in [P]$. Our goal is to find solutions $\{\chi_p\}_{p=1}^P$ that approach $\chi_p^* = \arg\min_{\chi} \phi_p(\chi)$. In this paper, we assume that m clients have P different data distributions, $D_1, D_2, \ldots, D_P$, and are divided into P disjoint clusters, $C_1^*, C_2^*, \ldots, C_P^*$, and every client $i \in C_p^*$ owns n i.i.d. data point $y^{i,1}, \ldots, y^{i,n}$. In particular, we define the i-th client's loss function

$$\phi_i(\chi_p; Y) = \frac{1}{|Y|} \sum_{y \in Y} f(\chi_p; y), \tag{2}$$

where $Y \subseteq \{y^{i,1}, \ldots, y^{i,n}\}$ represents the set of data points by the i-th client. This loss function is defined as $f(\chi_p; y) \colon \chi \to \mathbb{R}^d$ with respect to the data point y. $\phi_i(\chi_p)$ is the local lost function. The CFL-MGD learning process is as follows: We adopted two optimization schemes, i.e., average gradient and aggregate model parameters. Foremost, we set initial values for $\mathbf{u}_i^{(0)}$, $\chi_i^{(0)}$ and $\chi_p^{(0)}$, $p \in [P]$.

Local Updating:

(a) For gradient averaging, we first compute the (stochastic) gradient $\nabla \phi_i(\chi_p^{(t)})$ and update the momentum

$$\mathbf{u}_i^{(t)} = \beta \mathbf{u}_i^{(t-1)} + \nabla \phi_i(\chi_p^{(t-1)}). \tag{3}$$

(b) For model averaging, local updates are performed in parallel on each client:

$$\begin{aligned} \mathbf{u}_i^{(t)} &= \beta \mathbf{u}_i^{(t-1)} + \nabla \phi_i(\chi_i^{(t-1)}) \\ \chi_i^{(t)} &= \chi_i^{(t-1)} - \eta \mathbf{u}_i^{(t)}. \end{aligned} \tag{4}$$

According to (4), node i optimizes its loss function $\phi_i(\chi)$ by performing MGD. The reason for using MGD is to improve the convergence of the algorithm.

Global aggregation: client i transmits $\mathbf{u}_i^{(t)}$, $\chi_i^{(t)}$ and $\nabla\phi_i(\chi_p^{(t)})$ to the central server, which averages the received parameters from m clients to obtain the cluster parameters, $\mathbf{u}_p^{(t)}$ and $\chi_p^{(t)}$, respectively. There are two approaches to this aggregation rule:

(a) Gradient averaging:

$$\begin{cases} \chi_p^{(t)} = \chi_p^{(t-1)} - \frac{\eta}{m} \sum\limits_{i \in M_t} r_{i,p} \mathbf{u}_i^{(t)} \\ \mathbf{u}_p^{(t)} = \frac{\sum\limits_{i \in M_t} r_{i,p} \mathbf{u}_i^{(t)}}{\sum\limits_{i \in M_t} r_{i,p}}. \end{cases} \tag{5}$$

(b) Model averaging:

$$\begin{cases} \chi_p^{(t)} = \sum\limits_{i \in M_t} r_{i,p} \chi_i^{(t)} / \sum\limits_{i \in M_t} r_{i,p} \\ \mathbf{u}_p^{(t)} = \frac{\sum\limits_{i \in M_t} r_{i,p} \mathbf{u}_i^{(t)}}{\sum\limits_{i \in M_t} r_{i,p}}. \end{cases} \tag{6}$$

Here, $r_{i,p}$ is the one-hot encoding vector. The central server sends the aggregated cluster parameters $\mathbf{u}_p^{(t)}$ to i with $r_{i,p} = 1$, and broadcasts the parameters $\chi_p^{(t)}$ for the next iteration. If we let $\chi_i^{(-1)} = \chi_i^{(0)}$, then (4) can be equivalently written as the following single variable version

$$\chi_i^{(t)} = \chi_i^{(t-1)} - \eta \nabla \phi_i(\chi_i^{(t-1)}) + \beta(\chi_i^{(t-1)} - \chi_i^{(t-2)}), \tag{7}$$

where the term $\beta(\chi_i^{(t-1)} - \chi_i^{(t-2)})$ is habitually referred to as Polyak's momentum.

4. CFL-MGD Algorithm

A detailed introduction of the algorithm is given in this section, namely clustered federated learning based on momentum gradient descent (CFL-MGD). The primary idea is to cross-iterate the minimization loss function and evaluate the category of clusters. There are two processes: initially, the server broadcasts P parameters and randomly selects participating clients. Next, each client calculates and selects the model parameters that minimize the local loss function

$$\hat{p} = \arg\min_{p \in [P]} \phi_i(\chi_p^{(t)}). \tag{8}$$

For gradient averaging, each client calculates the gradient $\hat{\nabla}\phi_i(\chi_{\hat{p}}^{(t)})$ and updates the momentum buffer. For model averaging, each client performs MGD using (4). Ultimately, the solution of each cluster is updated in parallel and the next iteration continues until the end of the loop, using (5) or (6). The algorithm is formally expressed in Algorithm 1.

Algorithm 1 Clustered federated learning based on momentum gradient descent (CFL-MGD).

Input: Total of cluster P, step size η, β, $p \in [P]$, initialization $\chi_p^{(0)}$, $\mathbf{u}_i^{(0)} = \mathbf{0}$

for $t = 1, 2, \ldots, T$ **do**
  center server: broadcast $\chi_p^{(t)}$, $p \in [P]$
  $M_t \leftarrow$ Randomly select participating clients
  for client $i \in M_t$ in parallel **do**
    evaluate cluster categories: $\hat{p} = \arg\min_{p \in [P]} \phi_i(\chi_p^{(t)})$
    define vector $r_i = \{r_{i,p}\}_{p=1}^{P}$ with $r_{i,p} = 1\{p = \hat{p}\}$
    option I (gradient averaging):
      compute gradient: $g_i^{(t)} = \hat{\nabla}\phi_i(\chi_{\hat{p}}^{(t)})$
      update momentum: $\mathbf{u}_i^{(t)} = \beta \mathbf{u}_i^{(t-1)} + g_i^{(t-1)}$
      reverse back r_i, $g_i^{(t)}$ and $\mathbf{u}_i^{(t)}$ to the center server
    option II (model averaging):
      $\tilde{\chi}_i^{(t)}, \mathbf{u}_i^{(t)}$=LocalUpdate$(\chi_{\hat{p}}^{(t)}, \mathbf{u}_p^{(t)}, \eta, H)$, reverse back r_i, $\tilde{\chi}_i^{(t)}$ and $\mathbf{u}_i^{(t)}$ to the center server
  end for
  center server:
  option I (gradient averaging):

$$\begin{cases} \chi_p^{(t)} = \chi_p^{(t-1)} - \frac{\eta}{m} \sum\limits_{i \in M_t} r_{i,p} \mathbf{u}_i^{(t)} \\ \mathbf{u}_p^{(t)} = \frac{\sum\limits_{i \in M_t} r_{i,p} \mathbf{u}_i^{(t)}}{\sum\limits_{i \in M_t} r_{i,p}} \end{cases} \quad \text{send } \mathbf{u}_p^{(t)} \text{ to } i \text{ with } r_{i,p} = 1 \text{ and set } \mathbf{u}_i^{(t)} = \mathbf{u}_p^{(t)}$$

  option II (model averaging):

$$\begin{cases} \chi_p^{(t)} = \sum\limits_{i \in M_t} r_{i,p} \tilde{\chi}_i^{(t)} / \sum\limits_{i \in M_t} r_{i,p} \\ \mathbf{u}_p^{(t)} = \sum\limits_{i \in M_t} r_{i,p} \mathbf{u}_i^{(t)} / \sum\limits_{i \in M_t} r_{i,p} \end{cases} \quad \text{send } \mathbf{u}_p^{(t)} \text{ to } i \text{ with } r_{i,p} = 1$$

end for
return $\chi_p^{(T)}$, $p \in [P]$
LocalUpdate$(\tilde{\chi}_i^{(0)}, \mathbf{u}_i^{(0)}, \eta, H)$ at the i-th client
for $s = 1, 2, \ldots, H$ **do**
  momentum gradient descent: $\begin{cases} \tilde{\chi}_i^{(s)} = \tilde{\chi}_i^{(s-1)} - \eta \mathbf{u}_i^{(s)} \\ \mathbf{u}_i^{(s)} = \beta \mathbf{u}_i^{(s-1)} + \nabla \phi_i(\tilde{\chi}_i^{(s-1)}) \end{cases}$
end for
return $\tilde{\chi}_i^{(H)}$, $\mathbf{u}_i^{(H)}$

5. Theoretical Guarantees

In this section, we provide a theoretical analysis of the CFL-MGD algorithm by adopting the strategies of gradient averaging and model averaging. We first assume that all clients participate in each iteration. Moreover, we utilize resampling techniques [47] in our theoretical guarantees to reduce the dependency between the category evaluation and computed gradient. In particular, we add momentum to each client to speed up convergence. For the resampling techniques, if the aggregate number of iterations is T, we divide the n data points owned by every client into $2T$ disjoint subsets. Thus, $n' = \frac{n}{2T}$ represents the number of data points in each iteration used to compute the (stochastic) gradient or evaluate cluster categories. For the i-th client, we use the subsets $\hat{Y}_i^{(1)}, \ldots, \hat{Y}_i^{(T)}$ to evaluate cluster categories and use $Y_i^{(1)}, \ldots, Y_i^{(T)}$ to compute the (stochastic) gradient. In particular, for the i-th client in the t-th iteration, we employ $\hat{Y}_i^{(t)}$ to evaluate the cluster categories and employ $Y_i^{(t)}$ to compute the (stochastic) gradient. The advantage of this is that we use fresh

data in each iteration and, accordingly, reduce dependencies between category evaluation and computed gradient.

Naturally, the gradient update rule for the p-th cluster is

$$\begin{cases} C_p^{(t)} = \{i \in [m] : \hat{p} = \arg\min \phi_i(\chi_p^{(t)}; \hat{Y}_i^{(t)})\} \\ \chi_p^{(t)} = \chi_p^{(t-1)} - \frac{\eta}{m} \sum\limits_{i \in M_t} r_{i,p} \mathbf{u}_i^{(t)} \\ \mathbf{u}_i^{(t)} = \beta \mathbf{u}_i^{(t-1)} + g_i^{(t-1)}, \end{cases} \tag{9}$$

where $C_p^{(t)}$ presents the set of clients whose cluster category is estimated to be p at the t-th iteration.

Specifically, the model update rule for the p-th cluster is

$$\begin{cases} C_p^{(t)} = \{i \in [m] : \hat{p} = \arg\min \phi_i(\chi_p^{(t)}; \hat{Y}_i^{(t)})\} \\ \chi_p^{(t+1)} = \sum\limits_{i \in M_t} r_{i,p} \tilde{\chi}_i^{(t)} / \sum\limits_{i \in M_t} r_{i,p} \\ \tilde{\chi}_i^{(s)} = \tilde{\chi}_i^{(s-1)} - \eta \mathbf{u}_i^{(s)} \\ \mathbf{u}_i^{(s)} = \beta \mathbf{u}_i^{(s-1)} + \nabla \phi_i(\tilde{\chi}_i^{(s-1)}), \end{cases} \tag{10}$$

where $\mathbf{u}$ represents accumulated momentum, β is the momentum coefficient, η is the learning rate, and $\tilde{\chi}_i^{(0)}$ represents the model parameter $\chi_p^{(t)}$ selected by client i at time t from the server.

The following is the convergence guarantee of CFL-MGD. To further analyze our algorithm, we suppose that the cluster loss function $\phi_p(\chi)$ satisfies the below assumptions, and that we adopt at least as much data per iteration per cluster as the dimension of the parameter space, i.e., $lmn' \gtrsim d$. In addition, we define $l_p := |C_p^*|/m$ as the proportion of clients pertaining to the p-th cluster. In particular, we set $l := \min\{l_1, \ldots, l_P\}$. Moreover, we define $\Delta := \min_{p \neq p'} \|\chi_p^* - \chi_{p'}^*\|$. In particular, the assumption on Δ is to make sure that the iterates maintain an ℓ_2 ball around χ_p^*. In this paper, we require the following assumptions:

Assumption 1. *(Smoothness): For all $p \in [P]$, $\phi_p(\chi)$ is L-smooth, i.e., $\|\nabla \phi_p(\chi) - \nabla \phi_p(\chi')\| \leq L\|\chi - \chi'\|$.*

Assumption 2. *(Strong convexity): For all $p \in [P]$, $\phi_p(\chi)$ is μ-strongly convex, i.e., $\phi_p(\chi') \geq \phi_p(\chi) + \langle \nabla \phi_p(\chi), \chi' - \chi \rangle + \frac{\mu}{2}\|\chi' - \chi\|^2$.*

Assumption 3. *(Boundedness): For every χ and $p \in [P]$, $\mathbb{E}_{y \sim D_p}[(\phi(\chi; y) - \phi_p(\chi))^2] \leq \sigma^2$ and $\mathbb{E}_{y \sim D_p}[\|\nabla \phi(\chi; y) - \nabla \phi_p(\chi)\|^2] \leq v^2$.*

Assumption 4. *(Initialization): The initialization of parameters $\chi_p^{(0)}$ satisfies $\|\chi_p^{(0)} - \chi_p^*\| \leq \frac{1}{4}\sqrt{\frac{\mu}{L}}\Delta, \forall p \in [P]$, $n' \gtrsim \frac{p\sigma^2}{\mu^2 \Delta^4}$. Moreover, we also assume $\max_{p \in [P]} \|\chi_p^*\| \lesssim 1$.*

Assumption 5. *(Unbiasedness): The gradient estimator is unbiased, i.e., $\mathbb{E}[g_i] = \hat{\nabla} \phi_i(\chi_{\hat{p}}^{(t)})$.*

For the gradient averaging strategy, we provide the analysis of our algorithm based on the above assumptions. Moreover, we assume that T iterations are performed, obtain parameter vectors $\chi_p^{(T)}$ close to the truth parameters χ_p^*, and prove that $\chi_p^{(T)}$ converges to χ_p^* at an exponential rate.

Theorem 1. *Suppose Assumptions 1–5 hold. We choose $\eta = 1/L$, with a probability of at least $1 - \lambda$, for any $\lambda \in (0, 1)$ and $\lambda = 1 - \lambda_0 - P\lambda_1 - 2\lambda_2 - \lambda_3 - P\lambda_4 - 4\exp(-clm)$.*

After $T = \frac{8L}{l\mu}\log(\frac{\Delta}{2\rho})$ *parallel iterations, constants* $c_1, c_2, c_3, c_4, c_5, c_6 > 0$ *exist, and we have for all* $p \in [P]$

$$\|\chi_p^{(T)} - \chi_p^*\| \leq \rho, \tag{11}$$

where $\rho = \frac{16L}{\mu p}\rho_0 \lesssim \tilde{\mathcal{O}}(\frac{1}{\sqrt{mn'}} + \frac{1}{n'} + \frac{1}{n'\sqrt{m}})$, *and* $\rho_0 = \frac{2v}{\lambda_0 L\sqrt{lmn'}} + c_1\frac{\sigma^2}{\mu^2\lambda_2\Delta^4 n'} + c_2\frac{v\sigma\sqrt{P}}{\lambda_1\mu L\Delta^2 n'\sqrt{m\lambda_2}} + c_3\frac{\beta v\sqrt{l}}{\lambda_3(1-\beta)L\sqrt{mn'}} + c_4 l + c_5\frac{\beta\sigma^2}{\mu^2\lambda_2\Delta^4 n'} + c_6\frac{\beta\sigma v\sqrt{P}}{\lambda_4\mu L\Delta^2 n'(1-\beta)\sqrt{m\lambda_2}}$.

We prove Theorem 1 in Appendix B.

Remark 1. *To better understand the results, let us pay close attention to m and n and take the remaining quantities as constant terms. Due to* $n = 2n'T$, *the convergence rate can be written as* $\tilde{\mathcal{O}}(\frac{1}{\sqrt{mn}} + \frac{1}{n})$. $\frac{1}{\sqrt{mn}}$ *is the optimal rate if we know the class of clusters. Compared to ([23], Theorem 2) of the statistical rate in strongly convex models* $\tilde{\mathcal{O}}(\frac{1}{\sqrt{mn}} + \frac{1}{n})$, *the CFL-MGD algorithm achieves a similar rate of convergence.*

Specifically, for the model averaging strategy, we conducted the following analysis on the convergence of the proposed algorithm. Assuming the model parameter uploaded by the i-th client is $\tilde{\chi}_i^{(t)}$, the convergence of this algorithm can be shown by studying the following formula.

$$\begin{aligned}\|\chi_p^{(t+1)} - \chi_p^*\| &= \left\|\frac{1}{|C_p|}\sum_{i\in C_p}\chi_i^{(t)} - \frac{\eta}{|C_p|}\sum_{i\in C_p}\nabla\phi_i(\chi_i^{(t-1)}) + \frac{\beta}{|C_p|}\sum_{i\in C_p}(\chi_i^{(t-1)} - \chi_i^{(t-2)}) - \chi_p^*\right\| \\ &\leq \frac{1}{|C_p|}\sum_{i\in C_p}\|\chi_i^{(t)} - \chi_p^*\| + \frac{\eta}{|C_p|}\sum_{i\in C_p}\|\nabla\phi_i(\chi_i^{(t-1)})\| + \frac{\beta}{|C_p|}\sum_{i\in C_p}\|\chi_i^{(t-1)} - \chi_i^{(t-2)}\|.\end{aligned} \tag{12}$$

Next, we can use the Lipschitz continuity of $\nabla\phi_i$:

$$\begin{aligned}\|\chi_p^{(t+1)} - \chi_p^*\| \leq& \frac{1}{|C_p|}\sum_{i\in C_p}\|\chi_i^{(t)} - \chi_p^*\| + \frac{\eta L}{|C_p|}\sum_{i\in C_p}\|\chi_i^{(t-1)} - \chi_p^*\| \\ &+ \frac{\beta}{|C_p|}\sum_{i\in C_p}\|\chi_i^{(t-1)} - \chi_i^{(t-2)}\|.\end{aligned} \tag{13}$$

We can simplify the second term by using the recursive definition of χ_i:

$$\begin{aligned}\|\chi_p^{(t+1)} - \chi_p^*\| \leq& \frac{1}{|C_p|}\sum_{i\in C_p}\|\chi_i^{(t)} - \chi_p^*\| + \frac{\eta L}{|C_p|}\sum_{i\in C_p}\left\|\chi_p^{(t-1)} - \frac{\eta}{|C_p|}\sum_{j\in C_p}\nabla\phi_j(\chi_j^{(t-2)})\right. \\ &\left. + \frac{\beta}{|C_p|}\sum_{j\in C_p}(\chi_j^{(t-2)} - \chi_j^{(t-3)}) - \chi_p^*\right\| \\ =& \left(1 + \frac{\eta L}{|C_p|}\right)\frac{1}{|C_p|}\sum_{i\in C_p}\|\chi_i^{(t)} - \chi_p^*\| + \frac{\beta}{|C_p|}\sum_{i\in C_p}\|\chi_i^{(t-1)} - \chi_i^{(t-2)}\|.\end{aligned} \tag{14}$$

Finally, we can use $|C_p| \geq 1$ to simplify the expression:

$$\begin{aligned}\|\chi_p^{(t+1)} - \chi_p^*\| &\leq \left(1 + \frac{\eta L}{|C_p|}\right)\|\chi_p^{(t)} - \chi_p^*\| + \frac{\beta}{|C_p|}\sum_{i\in C_p}\|\chi_i^{(t-1)} - \chi_i^{(t-2)}\| \\ &= \left(1 + \frac{\eta L}{|C_p|}\right)\|\chi_p^{(t)} - \chi_p^*\| + \beta(1+\beta)\frac{\|\chi_p^{(t-1)} - \chi_p^{(t-2)}\|^2}{\|\chi_p^{(t-2)} - \chi_p^{(t-1)}\|^2}.\end{aligned} \tag{15}$$

By the definition of $\epsilon_t = \|\chi_p^{(t)} - \chi_p^*\|$, we can obtain

$$\epsilon_{t+1} \leq \left(1 + \frac{\eta L}{|C_p|}\right)\epsilon_t + \beta(1+\beta)\frac{\epsilon_{t-1}^2}{\epsilon_{t-2}^2}, \tag{16}$$

where $\epsilon_0 = \|\chi_p^{(0)} - \chi_p^*\|$ and $\epsilon_{-1} = \|\chi_p^{-1} - \chi_p^*\|$. For convenience, we define $M = \max\{\left(1 + \frac{\eta L}{|C_p|}\right), \beta(1+\beta)\}$, we can obtain

$$\begin{aligned}
\epsilon_{t+1} &\leq M\epsilon_t + \frac{\beta(1+\beta)}{M}\frac{\epsilon_{t-1}^2}{\epsilon_{t-2}^2} \\
&\leq M^2\epsilon_{t-1} + \frac{\beta(1+\beta)}{M}\frac{\epsilon_{t-2}^2}{\epsilon_{t-3}^2} + \frac{\beta(1+\beta)}{M}\frac{\epsilon_{t-1}^2}{\epsilon_{t-2}^2} \\
&\leq M^3\epsilon_{t-2} + \frac{\beta(1+\beta)}{M}\frac{\epsilon_{t-3}^2}{\epsilon_{t-4}^2} + \frac{\beta(1+\beta)}{M}\frac{\epsilon_{t-2}^2}{\epsilon_{t-3}^2} + \frac{\beta(1+\beta)}{M}\frac{\epsilon_{t-1}^2}{\epsilon_{t-2}^2} \\
&\vdots \\
&\leq M^{t+1}\epsilon_{-1} + \sum_{i=0}^{t-1}\left(\frac{\beta(1+\beta)}{M}\right)^{t-\tau}\frac{\epsilon_\tau^2}{\epsilon_{\tau-1}^2}.
\end{aligned} \tag{17}$$

If we take $t = \lfloor \log_2 \epsilon_0^{-1} \rfloor$, we have $\epsilon_{t+1} \leq \epsilon_t$, i.e., $\{\epsilon_t\}$ is monotonically decreasing.

$$\begin{aligned}
\epsilon_0 &\leq M^{t+1}\epsilon_{-1} + \sum_{\tau=0}^{t-1}\left(\frac{\beta(1+\beta)}{M}\right)^{t-\tau}\frac{\epsilon_i^2}{\epsilon_{\tau-1}^2} \\
&\leq M^{t+1}\epsilon_{-1} + \frac{\beta(1+\beta)}{M-1}\sum_{\tau=0}^{t-2}\left[\left(\frac{\epsilon_\tau}{\epsilon_{\tau-1}} - \frac{\beta}{M}\right)^2 - \frac{\beta^2}{M^2}\right]\epsilon_{\tau-1}^2 + \frac{\beta^2}{M-1}\epsilon_{t-1}^2 \\
&\leq \frac{M^{t+2}-1}{M-1}\epsilon_{-1} + \frac{\beta(1+\beta)}{(M-1)M^2}\sum_{\tau=0}^{t-2}\epsilon_\tau^2 + \frac{\beta^2}{M-1}\epsilon_{t-1}^2.
\end{aligned} \tag{18}$$

Here, we use inequality $\frac{(x-y)^2}{4} \leq \frac{x^2+y^2}{2}$ and $(a+b)^2 \leq 2(a^2+b^2)$. Since ϵ_t is monotonically decreasing, $\sum_{i=0}^{t-2}\epsilon_i^2 \leq t\epsilon_0^2$. Meanwhile, we have $t \leq \log_2 \frac{1}{\epsilon_0}$. Thus, the convergence rate can be derived as follows:

$$\epsilon_0 \leq \frac{2M^3}{\beta}\frac{1}{T+2}\left(\frac{\eta L}{|C_p|} + \beta(1+\beta)\right)\log_2\frac{1}{\epsilon_0} + \frac{M^{t+2}-1}{M-1}\epsilon_{-1}, \tag{19}$$

Because $\epsilon_0 \geq \epsilon_{-1}$, we can simplify the upper bound of the rate of convergence as follows:

$$\epsilon_0 \leq \frac{2M^3}{\beta}\frac{1}{T+2}\left(\frac{\eta L}{|C_p|} + \beta(1+\beta)\right)\log_2\frac{1}{\epsilon_0} + \frac{M^{T+3}}{M-1}\epsilon_0. \tag{20}$$

Therefore, we obtain the upper bound of $\|\chi_p^{(t+1)} - \chi_p^*\|$, which is a function of T.

We initially consider the following Lemma 1 prior to proving Theorem 1. Suppose that Assumption 4 is satisfied. We analyze the error probability of clients being classified into error clusters. Defining event $\xi_i^{p,p'}$ indicates that clients belonging to the p-th cluster C_p are divided into p'-th cluster $C_{p'}$, which means that client i is correctly classified as $\xi_i^{(p,p)}$. Therefore, we have

$$\xi_i^{1,p} = \left\{\phi_i(\chi_1; \hat{Y}_i \geq \phi_i(\chi_p; \hat{Y}_i)\right\}, \tag{21}$$

where $\hat{Y}_i$ is a collection of data points n' owned by client i to evaluate the cluster category.

Lemma 1. *We assume that client $i \in C_p^*$. Then, there exists a constant c_1 for arbitrary $p' \neq p$; we have*

$$\mathbb{P}(\xi_i^{p,p'}) \leq c_1 \frac{\sigma^2}{\Delta^4 \mu^2 n'}, \tag{22}$$

by union bound

$$\mathbb{P}(\xi_i^{p,p'}) \leq c_1 \frac{P\sigma^2}{\Delta^4 \mu^2 n'}. \tag{23}$$

Proof of Lemma 1. Shorthand $\phi_i(\chi) := \phi_i(\chi; \hat{Y}_i)$. Then,

$$\mathbb{P}(\xi_i^{(1,p)}) \leq \mathbb{P}(\phi_i(\chi_1) > \delta) + \mathbb{P}(\phi_i(\chi_p) \leq \delta), \tag{24}$$

for arbitrary $\delta \geq 0$, we select $\delta = \frac{\phi^1(\chi_1)+\phi^1(\chi_p)}{2}$. We have

$$\begin{aligned} &\mathbb{P}(\phi_i(\chi_1) > \delta) \\ =&\mathbb{P}\left(\phi_i(\chi_1) > \frac{\phi^1(\chi_1) + \phi^1(\chi_p)}{2}\right) \\ =&\mathbb{P}\left(\phi_i(\chi_1) - \phi^1(\chi_1) > \frac{\phi^1(\chi_p) - \phi^1(\chi_1)}{2}\right). \end{aligned} \tag{25}$$

Similarly, we acquire $\mathbb{P}(\phi_i(\chi_p) \leq \delta) = \mathbb{P}\left(\phi_i(\chi_p) - \phi^1(\chi_p) \leq -\frac{\phi^1(\chi_p)-\phi^1(\chi_1)}{2}\right)$.

According to Assumption 2, we obtain

$$\phi^1(\chi_p) \geq \phi^1(\chi_1^*) + \frac{\mu}{2}\|\chi_p - \chi_1^*\|^2 \geq \phi^1(\chi_1^*) + \frac{9\mu}{32}\Delta^2, \tag{26}$$

where the second inequality is satisfied by $\|\chi_p - \chi_1^*\| \geq \Delta - \frac{1}{4}\sqrt{\frac{\mu}{L}}\Delta \geq \frac{3}{4}\Delta$. According to Assumptions 1 and 4, we know that

$$\phi^1(\chi_1) \leq \phi^1(\chi_1^*) + \frac{L}{2}\|\chi_1 - \chi_1^*\| \leq \phi^1(\chi_1^*) + \frac{\mu}{32}\Delta^2. \tag{27}$$

Combining (26) and (27) yields

$$\phi^1(\chi_p) - \phi^1(\chi_1) \geq \frac{\mu}{4}\Delta^2. \tag{28}$$

According to Chebyshev's inequality, we acquire

$$\mathbb{P}(\phi_i(\chi_1) > \delta) \leq \frac{64\sigma^2}{\Delta^4 \mu^2 n'}, \tag{29}$$

and

$$\mathbb{P}(\phi_i(\chi_p) \leq \delta) \leq \frac{64\sigma^2}{\Delta^4 \mu^2 n'}. \tag{30}$$

The proof is now complete. □

In order to prove Theorem 1, we first prove the following Lemma 2. We assume that at some iteration, we obtain a vector of parameters $\chi^{(t)}$.

Lemma 2. *Suppose Assumptions 1–5 hold. We choose* $\eta = 1/L$*, with a probability of at least* $1 - P\lambda_5$*, for any* $\lambda \in (0,1)$*. For all* $p \in [P]$*, we have*

$$\|\chi_p^{(t+1)} - \chi_p^*\| \leq (1 - \frac{l\mu}{8L})\|\chi_p^{(t)} - \chi_p^*\| + \rho_0. \tag{31}$$

Proof of Lemma 2. Without loss of generality, we focus on the first cluster

$$\begin{aligned} \chi_1^{(t+1)} =& \chi_1^{(t)} - \frac{\eta}{m}\sum_{i\in C_1}(\beta \mathbf{u}_i^{(t)} + g_i^{(t)}) \\ =& \chi_1^{(t)} - \frac{\eta}{m}\sum_{i\in C_1} g_i^{(t)} + \beta(\chi_1^{(t)} - \chi_1^{(t-1)}). \end{aligned} \tag{32}$$

As for $\|\chi_1^{(t+1)} - \chi_1^*\|$, it is written in (33), where the last term uses the triangle inequality $\|\mathbf{a} - \mathbf{b}\| \leq \|\mathbf{a}\| + \|\mathbf{b}\|$.

$$\begin{aligned} \|\chi_1^{(t+1)} - \chi_1^*\| =& \left\|\chi_1^{(t)} - \frac{\eta}{m}\sum_{i\in C_1} g_i^{(t)} + \beta(\chi_1^{(t)} - \chi_1^{(t-1)}) - \chi_1^*\right\| \\ =& \left\|\chi_1^{(t)} - \frac{\eta}{m}\sum_{i\in C_1\cap C_1^*} g_i^{(t)} - \frac{\eta}{m}\sum_{i\in C_1\cap \overline{C_1^*}} g_i^{(t)} + \beta(\chi_1^{(t)} - \chi_1^{(t-1)}) - \chi_1^*\right\| \\ =& \left\|\chi_1^{(t)} - \chi_1^* - \frac{\eta}{m}\sum_{i\in C_1\cap C_1^*} g_i^{(t)} - \frac{\eta}{m}\sum_{i\in C_1\cap \overline{C_1^*}} g_i^{(t)} - \frac{\beta\eta}{m}\sum_{i\in C_1}\mathbf{u}_i^{(t)}\right\| \\ =& \left\|\chi_1^{(t)} - \chi_1^* - \frac{\eta}{m}\sum_{i\in C_1\cap C_1^*} g_i^{(t)} - \left(\frac{\eta}{m}\sum_{i\in C_1\cap \overline{C_1^*}} g_i^{(t)} + \frac{\beta\eta}{m}\sum_{i\in C_1}\mathbf{u}_i^{(t)}\right)\right\| \\ =& \left\|\underbrace{\chi_1^{(t)} - \chi_1^* - \frac{\eta}{m}\sum_{i\in C_1\cap C_1^*} g_i^{(t)}}_{T_1} - \underbrace{\left(\frac{\eta}{m}\sum_{i\in C_1\cap \overline{C_1^*}} g_i^{(t)}\right)}_{T_2} + \underbrace{\frac{\beta\eta}{m}\sum_{i\in C_1}\mathbf{u}_i^{(t)}}_{T_3}\right\| \\ \leq& \|T_1\| + \|T_2\| + \|T_3\|. \end{aligned} \tag{33}$$

In the following part, we analyze the upper bounds of $\|T_1\|$, $\|T_2\|$, and $\|T_3\|$, respectively. Analysis of the $\|T_1\|$:

$$T_1 = \underbrace{\chi_1^{(t)} - \chi_1^* - \hat{\eta}\nabla\phi^1(\chi_1)}_{T_{11}} + \underbrace{\hat{\eta}(\nabla\phi^1(\chi_1) - \frac{1}{|C_1\cap C_1^*|}\sum_{i\in C_1\cap C_1^*} g_i^{(t)})}_{T_{12}}, \tag{34}$$

where $\hat{\eta} := \frac{\eta}{m}|C_1 \cap C_1^*|$. We obtain the following with a probability of at least $1 - \lambda_0 - 2\exp(-clm)$,

$$\|T_1\| \leq (1 - \frac{l\mu}{8L})\|\chi_1^{(t)} - \chi_1^*\| + \frac{2v}{\lambda_0 L\sqrt{lmn'}}. \tag{35}$$

Analysis of the $\|T_2\|$: We define $T_{2p} = \sum_{i\in C_1\cap C_p^*} g_i^{(t)}$, and $p \geq 2$. So $T_2 = \frac{\eta}{m}\sum_{p=2}^{P} T_{2p}$. With a probability of at least $1 - P\lambda_1 - \lambda_2$, we have

$$\|T_2\| \leq c_1\frac{\sigma^2}{\mu^2\lambda_2\Delta^4 n'} + c_2\frac{v\sigma\sqrt{P}}{\lambda_1\mu L\Delta^2 n'\sqrt{m\lambda_2}}. \tag{36}$$

Analysis of the $\|T_3\|$:

$$\begin{aligned} T_3 =& \frac{\beta\eta}{m}\sum_{i\in C_1}\mathbf{u}_i^{(t)} \\ =& \frac{\beta\eta}{m}\left(\sum_{i\in C_1\cap C_1^*}\mathbf{u}_i^{(t)}+\sum_{i\in C_1\cap\overline{C_1^*}}\mathbf{u}_i^{(t)}\right) \\ =& \underbrace{\frac{\eta\beta}{m}\sum_{i\in C_1\cap C_1^*}\mathbf{u}_i^{(t)}}_{T_{31}}+\underbrace{\frac{\beta\eta}{m}\sum_{i\in C_1\cap\overline{C_1^*}}\mathbf{u}_i^{(t)}}_{T_{32}}. \end{aligned} \tag{37}$$

With a probability of at least $1-\lambda_2-\lambda_3-P\lambda_4-2\exp(-clm)$, we have

$$\|T_3\| \le c_3\frac{\beta v\sqrt{lt}}{\lambda_3(1-\beta)L\sqrt{mn'}}+c_4 l+c_5\frac{\beta\sigma^2}{\mu^2\lambda_2\Delta^4 n'}+c_6\frac{\beta\sigma v\sqrt{Pt}}{\lambda_4\mu L\Delta^2 n'(1-\beta)\sqrt{m\lambda_2}}. \tag{38}$$

Submitting (35), (36), and (38) into (33), and using the union bound completes the proof. □

We give detailed proof of the upper bounds of $\|T_1\|$, $\|T_2\|$, and $\|T_3\|$ in Appendix A.

6. Simulation and Analysis

In this section, we will use the CIFAR-10 dataset [48] on the convolution neural network and MNIST dataset [49] on a fully connected neural network to verify our theoretical analysis. In the experiments, we compare our algorithm with IFCA algorithms [23], FedAvg [8], and FedBCD [50], and the experimental results show that our algorithm is more efficient and converges faster. Moreover, we relax the initialization requirements and still achieve a good convergence rate.

6.1. Datasets

CIFAR-10 and MNIST datasets were used to construct the experimental environment. To simulate different clients maintaining different data, we rotated the images. The CIFAR-10 dataset includes 60,000 color images, with 50,000 for training and 10,000 for testing. It is divided into 10 categories, with 6000 images per category. We enlarged the dataset by rotating the images by 0 and 180 degrees, resulting in 2 clusters ($P=2$). We assumed m clients and divided each client to contain n images of the same rotation operation to conform to $mn=60{,}000P$. The test sets were also equally divided into $m_{test}=10{,}000P/n$ clients. For the MNIST dataset, we performed the same operation but divided it into 4 clusters ($P=4$) by rotating the images by 0, 90, 180, and 270 degrees. The rotation operation is an effective method to enlarge datasets and is frequently used in clustered FL.

6.2. Neural Network Model

In our paper, we use two neural network models. One is a convolutional neural network [51] and the other is a fully connected neural network [52]. The convolutional neural network (CNN) is constructed from the bottom up. First, the input images go through a convolution layer, and then the resulting information is processed through pooling (Max pooling is used here). Then, after the same processing, the information obtained in the second step is transmitted to the fully connected neural layer consisting of two layers, which is also a general two-layer neural network. Finally, a classifier is connected for classification and prediction. The fully connected neural network is a multi-layer perceptron (MLP) [52], which is a network of multi-layer neurons. Non-linear activation functions are needed between layers, and there must be a hidden layer that conceals both inputs and outputs. Additionally, a high degree of connectivity is determined by the synaptic weight of the network.

In this paper, the convolutional neural network contains two convolution layers and two fully connected layers. This kind of neural network model is universal. Accordingly, another common one is a fully connected neural network model, which contains the ReLU activation function adopted in the experimental setting of the MNIST dataset.

6.3. Results Analysis

We compare our algorithm with IFCA [23], FedAvg [8], and FedBCD [50] algorithms. To facilitate the representation of the legend, let us abbreviate CFL-MGD as MCFL. For CIFAR-10 experiments, we divided the $P = 2$ clusters with $m = 200$ clients and $n = 500$ data. In addition, we set the participation rate to 0.1, the step size decay to 0.99, and chose a learning rate of 0.01 and momentum of 0.9. For MNIST experiments, we divided the $P = 4$ clusters with $m = 2400$ clients and $n = 100$ data, with a learning rate of 0.1. For FedAvg, the algorithm learns a single global model from data owned by all clients and ignores the identities of the clusters. In the IFCA scheme, the aggregation step in Algorithm becomes $\chi_p^{(t)} = \sum_{i \in M_t} r_{i,p} \tilde{\chi}_i^{(t)} / \sum_{i \in M_t} r_{i,p}$. The CFL-MGD algorithm is similar to the IFCA algorithm. However, the difference is that MGD is used in the local update process to accelerate convergence.

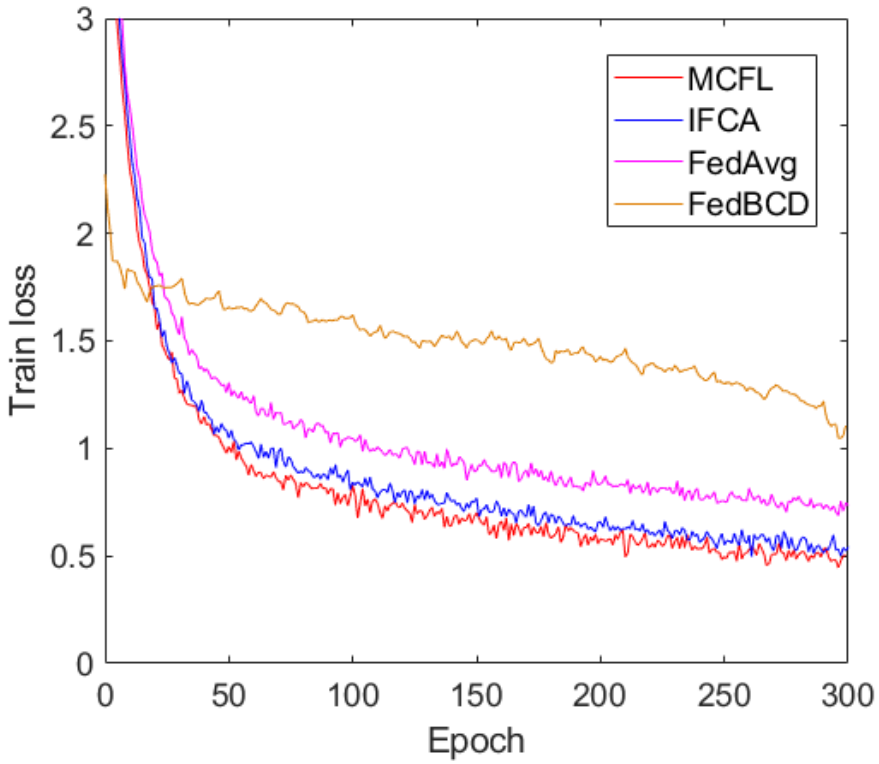
The experimental results on the CIFAR-10 dataset are shown in Figure 3. It can be observed from Figure 3a that our algorithm achieves smaller loss values compared to the other three algorithms. Figure 3b shows that although our algorithm is only slightly better than IFCA in terms of test accuracy, it reaches stability earlier and changes faster in the first 100 rounds. This faster convergence speed is due to the addition of the momentum term, which reduces the amplitude of the oscillation. Since IFCA performs stochastic gradient descent locally, the performance gain due to the momentum will vanish. Compared with the remaining two algorithms, it is clear that the MCFL algorithm proposed in this paper outperforms both FedAvg and FedBCD in terms of both training loss and test accuracy. During the execution of the MCFL algorithm, it gradually discovers the underlying cluster categories of participating clients, and after identifying the correct cluster, training and testing each model with the same distribution of data leads to better accuracy. FedAvg performs worse than the proposed algorithm because it attempts to match all data from different distributions and does not provide personalized predictions. FedBCD performs worse than MCFL due to the multiple local computations.

The experimental results on the MNIST dataset are shown in Figure 4. It can be observed from Figure 4a that the training loss function curves of all algorithms gradually converge as the number of communication rounds increases. Similarly, it can be observed from Figure 4b that the test accuracy curve gradually rises as the number of communication rounds increases until iteration convergence. It is evident that the MCFL algorithm proposed in this paper is far more effective than the other three algorithms in terms of both training loss and test accuracy. On the other hand, in the first 100 rounds, the MCFL loss function curve decreases much faster than the other three algorithms and converges to a fixed point earlier. Therefore, the effectiveness of MCFL is verified. Furthermore, we developed a more detailed analysis of our experimental results, as shown in Table 2.

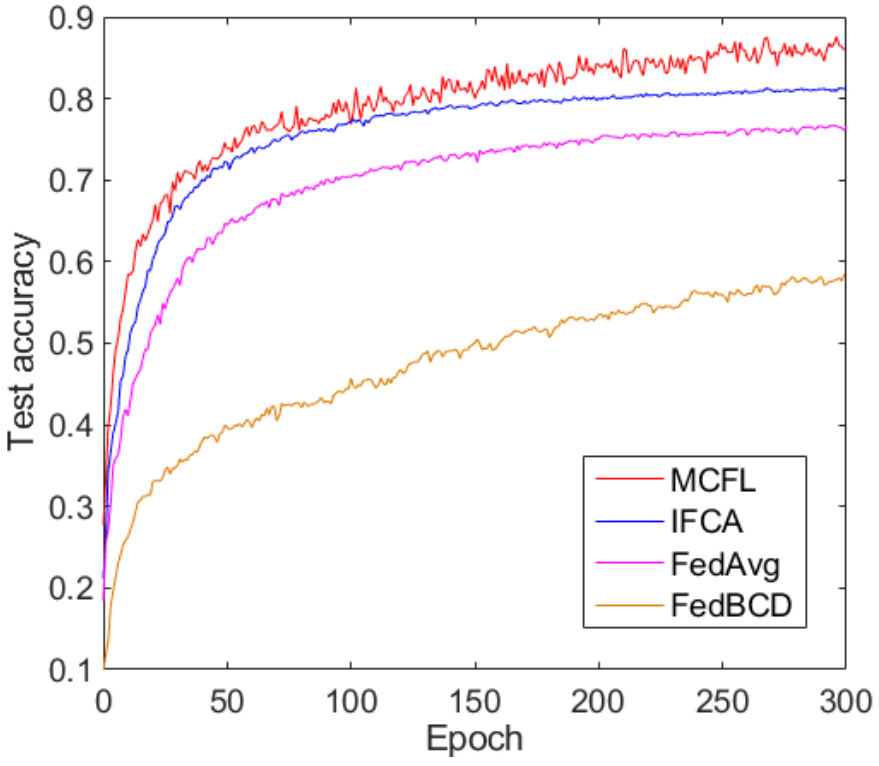
Table 2. Test accuracy (%) on CIFAR-10 and MNIST datasets (epoch = 300).

	CIFAR-10	**MNIST**
MCFL [1]	80.43 ± 0.14	96.05 ± 0.12
IFCA	75.53 ± 0.90	92.60 ± 0.27
FedAvg	69.52 ± 1.09	89.84 ± 0.32
FedBCD	47.31 ± 1.50	70.06 ± 0.92

[1] To facilitate the representation of the legend, let us abbreviate CFL-MGD as MCFL.

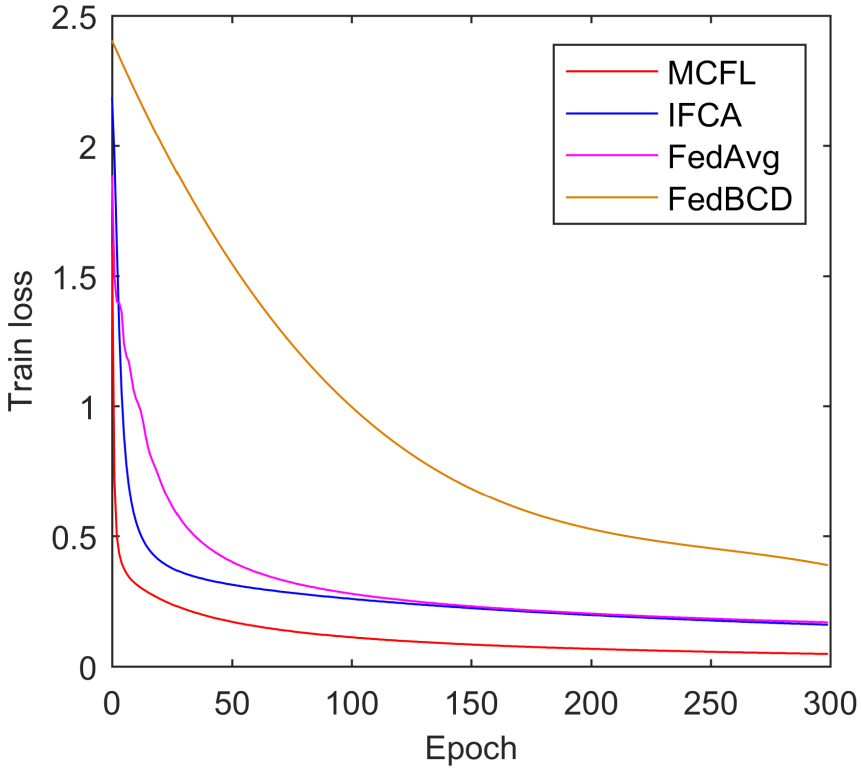


(**a**)

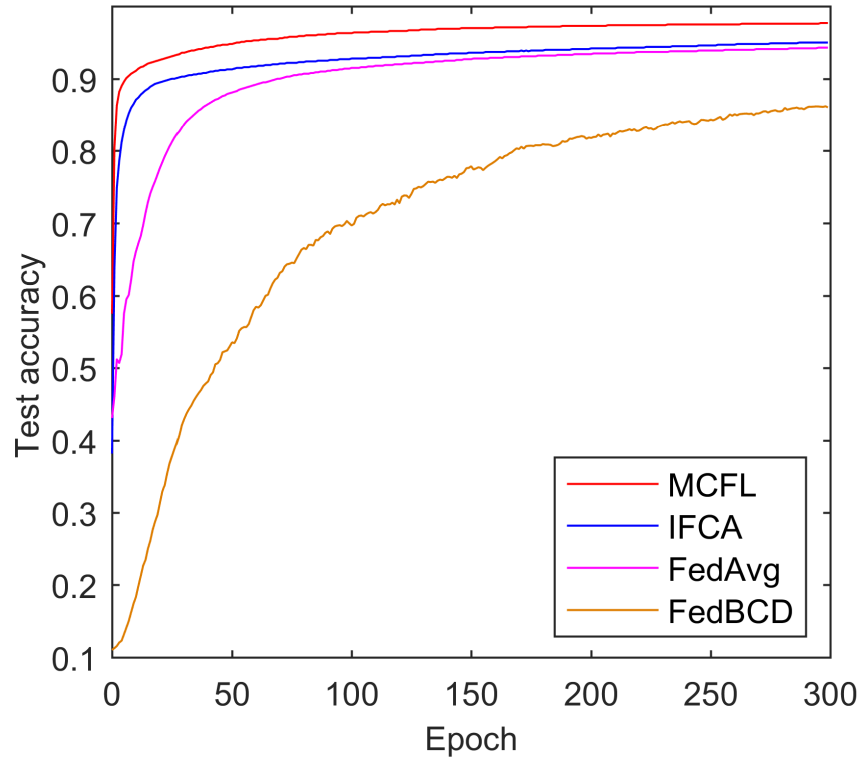


(**b**)

Figure 3. Test accuracy and training loss for different algorithms on the CIFAR-10 dataset. For IFCA, FedAvg, and our algorithm, the experimental environment is a convolution neural network that contains two convolutional layers and two fully connected layers, and for the FedBCD algorithm, the experimental environment is a deeper ResNet-20 model [53]. (**a**) Train loss vs. epoch; (**b**) test accuracy vs. epoch.



(**a**)



(**b**)

Figure 4. Test accuracy and training loss for different algorithms on the MNIST dataset. For IFCA, FedAvg, and our algorithm, the experimental environment is a fully connected neural network that contains the ReLU activation function and a hidden layer, and the number of clusters, $P = 4$; for the FedBCD algorithm, the experimental environment is a three-layered neural network. (**a**) Train loss vs. epoch; (**b**) test accuracy vs. epoch.

7. Conclusions

In this paper, we propose clustering federated learning based on momentum gradient descent. It can divide clients into appropriate clusters according to the clustering method of loss function minimization. Each client in the same cluster updates the local model parameters by momentum gradient descent using their private data and considers the momentum term and clustering in each iteration. This approach solves the suboptimal result caused by data heterogeneity and accelerates the convergence of the algorithm. For the gradient averaging and model averaging methods proposed in the global aggregation stage, we show that their convergence rates are $\tilde{\mathcal{O}}(\frac{1}{\sqrt{mn}}+\frac{1}{n})$, where $n=2n'T$ and $\tilde{\mathcal{O}}(\frac{1}{T})$, respectively. Moreover, we verify that our CFL-MGD algorithm improves the test accuracies by 4.90% and 3.45% compared to IFCA on the CIFAR-10 and MNIST datasets, respectively. In terms of the convergence rate of the algorithm, more significant improvements can be achieved compared with the clustering federation learning baseline IFCA. However, one potential risk is that our algorithm still requires users to send an estimate of their cluster category to the central server. Therefore, there may still be privacy concerns during this step. In future work, heterogeneous federated learning privacy protection schemes for complex scenarios can be further explored.

Author Contributions: Conceptualization, X.Z., P.X., L.X., G.Z. and H.M.; methodology, X.Z. and P.X.; software, X.Z. and P.X.; validation, X.Z., P.X., L.X., G.Z. and H.M.; formal analysis, X.Z. and P.X.; investigation, P.X. and H.M.; resources, X.Z., P.X., L.X. and H.M.; data curation, X.Z., P.X., G.Z. and H.M.; writing—original draft preparation, X.Z., P.X., L.X., G.Z. and H.M.; writing—review and editing, X.Z., P.X., G.Z. and H.M.; visualization, X.Z., P.X., L.X., G.Z. and H.M.; supervision, X.Z. and P.X.; project administration, X.Z., P.X. and L.X.; funding acquisition, P.X. and L.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grants no. 61801171 and 62072158, in part by the Henan Province Science Fund for Distinguished Young Scholars (222300420006), and Program for Innovative Research Team in University of Henan Province (21IRTSTHN015), and in part by the Key Technologies R and D program of Henan Province under Grants no. 212102210168 and 222102210001.

Data Availability Statement: Not applicable.

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

Appendix A. Proof of Lemma 2

Bound $\|T_1\|$

$$\begin{aligned} T_1 =& \underbrace{\chi_1^{(t)} - \chi_1^* - \hat{\eta}\nabla\phi^1(\chi_1)}_{T_{11}} \\ &+ \underbrace{\hat{\eta}(\nabla\phi^1(\chi_1) - \frac{1}{|C_1 \cap C_1^*|}\sum_{i\in C_1\cap C_1^*} g_i^{(t)})}_{T_{12}}, \end{aligned} \tag{A1}$$

where $\hat{\eta} := \frac{\eta}{m}|C_1 \cap C_1^*|$. Because the $\phi_p(\chi)$ is μ-strongly convex and L-smooth functions, we know that when $\hat{\eta} \leq \frac{1}{L}$, we obtain

$$\begin{aligned} \|T_{11}\| =& \|\chi_1^{(t)} - \chi_1^* - \hat{\eta}\nabla\phi^1(\chi_1)\| \\ \leq& (1 - \frac{\hat{\eta}\mu L}{\mu + L})\|\chi_1^{(t)} - \chi_1^*\|. \end{aligned} \tag{A2}$$

$\mathbb{E}[\|T_{12}\|^2] = \frac{v^2}{n'|C_1 \cap C_1^*|}$ can be acquired by Assumption 3, i.e., $\mathbb{E}[\|T_{12}\|] \leq \frac{v}{\sqrt{n'|C_1 \cap C_1^*|}}$. According to Markov's inequality, for any λ_0, with probability of at least $1 - \lambda_0$,

$$\|T_{12}\| \leq \frac{v}{\lambda_0 \sqrt{n'|C_1 \cap C_1^*|}}. \tag{A3}$$

Next, we analyze $|C_1 \cap C_1^*|$. Using Lemma 1 and ([23], Theorem 1), we can obtain the probability that each client i is correctly classified, given by $\mathbb{P}(\xi_i) \geq \frac{1}{2}$. Therefore, we have

$$\mathbb{E}[|C_1 \cap C_1^*|] \geq \mathbb{E}[\frac{1}{2}|C_1^*|] = \frac{1}{2} l_1 m, \tag{A4}$$

where $|C_1^*| = l_1 m$. Since $|C_1 \cap C_1^*|$ is a sum of the Bernoulli random variables with a success probability of at least $\frac{1}{2}$, we have

$$\begin{aligned} \mathbb{P}(|C_1 \cap C_1^*| \leq \frac{1}{4} l_1 m) \leq & \mathbb{P}(\big||C_1 \cap C_1^*| - \mathbb{E}[|C_1 \cap C_1^*|]\big| \geq \frac{1}{4} l_1 m) \\ \leq & 2\exp(-clm), \end{aligned} \tag{A5}$$

where $l = \min\{l_1, l_2, \ldots, l_P\}$, and the final step obeys Hoeffding's inequality. Therefore, we have

$$\mathbb{P}(|C_1 \cap C_1^*| \geq \frac{1}{4} l_1 m) \geq 1 - 2\exp(-clm). \tag{A6}$$

Assuming $|C_1 \cap C_1^*| \geq \frac{1}{4} l_1 m$ and choosing $\eta = \frac{1}{L}$, we have $\hat{\eta} \leq \frac{1}{L}$ and $\hat{\eta} \geq \frac{p}{4L}$. Combining with the facts in (A2), we have

$$\|T_{11}\| \leq (1 - \frac{l\mu}{8L}) \|\chi_1^{(t)} - \chi_1^*\|. \tag{A7}$$

Ultimately, we combine (A3) and (A7) to acquire the probability of at least $1 - \lambda_0 - 2\exp(-clm)$,

$$\|T_1\| \leq (1 - \frac{l\mu}{8L}) \|\chi_1^{(t)} - \chi_1^*\| + \frac{2v}{\lambda_0 L \sqrt{lmn'}}. \tag{A8}$$

Bound $\|T_2\|$ We define $T_{2p} = \sum_{i \in C_1 \cap C_p^*} g_i^{(t)}$ and $p \geq 2$. Thus, $T_2 = \frac{\eta}{m} \sum_{p=2}^{P} T_{2p}$. We first analyze T_{2p}

$$T_{2p} = |C_1 \cap C_p^*| \nabla \phi_p(\chi_1) + \sum_{i \in C_1 \cap C_p^*} (g_i^{(t)} - \nabla \phi_p(\chi_1)). \tag{A9}$$

According to Assumption 1, we have

$$\|\nabla \phi_p(\chi_1) - \nabla \phi_p(\chi_p^*)\| \leq L \|\chi_1 - \chi_p^*\| \leq 3L, \tag{A10}$$

the second step follows the fact that $\|\chi_1 - \chi_p^*\| \leq \|\chi_1 - \chi_1^*\| + \|\chi_p^*\| + \|\chi_1^*\| \leq 1 + 1 + \frac{1}{4}\sqrt{\frac{\mu}{L}}\Delta \leq 3$, and the second inequality applies to Assumption 4. Next, we analyze

$$\mathbb{E}\left[\left\| \sum_{i \in C_1 \cap C_p^*} (g_i^{(t)} - \nabla \phi_p(\chi_1)) \right\|^2\right] \leq |C_1 \cap C_p^*| \frac{v^2}{n'}, \tag{A11}$$

which means $\mathbb{E}\left[\left\| \sum_{i \in C_1 \cap C_p^*} (g_i^{(t)} - \nabla \phi_p(\chi_1)) \right\|\right] \leq \sqrt{|C_1 \cap C_p^*|} \frac{v}{\sqrt{n'}}$. According to Markov's inequality, for any $\lambda_1 \in (0, 1)$, with a probability of at least $1 - \lambda_1$,

$$\left\| \sum_{i \in C_1 \cap C_p^*} (g_i^{(t)} - \nabla \phi_p(\chi_1)) \right\| \leq \sqrt{|C_1 \cap C_p^*|} \frac{v}{\lambda_1 \sqrt{n'}}. \tag{A12}$$

By combining (A10) and (A12), we can obtain a probability of at least $1 - \lambda_1$,

$$\|T_{2p}\| \leq 3L|C_1 \cap C_p^*| + \sqrt{|C_1 \cap C_p^*|} \frac{v}{\lambda_1 \sqrt{n'}}. \tag{A13}$$

Eventually, applying the union bound, we can obtain a probability of at least $1 - P\lambda_1$,

$$\|T_2\| \leq \frac{3L\eta}{m} |C_1 \cap \overline{C_1^*}| + \frac{\eta v \sqrt{P}}{\lambda_1 m \sqrt{n'}} \sqrt{|C_1 \cap \overline{C_1^*}|}. \tag{A14}$$

Then, we analyze $|C_1 \cap \overline{C_1^*}|$, according to Lemma 1, we have

$$\mathbb{E}[|C_1 \cap \overline{C_1^*}|] \leq c_1 \frac{m\sigma^2}{\mu^2 \Delta^4 n'}. \tag{A15}$$

According to Markov's inequality, for any $\lambda_2 \in (0, 1)$, we have

$$\mathbb{P}\left(|C_1 \cap \overline{C_1^*}| \leq c_1 \frac{m\sigma^2}{\mu^2 \lambda_2 \Delta^4 n'} \right) \leq 1 - \lambda_2. \tag{A16}$$

Combining (A14) and (A16), and with a probability of at least $1 - P\lambda_1 - \lambda_2$,

$$\|T_2\| \leq c_1 \frac{\sigma^2}{\mu^2 \lambda_2 \Delta^4 n'} + c_2 \frac{v\sigma\sqrt{P}}{\lambda_1 \mu L \Delta^2 n' \sqrt{m\lambda_2}}. \tag{A17}$$

Bounding $\|T_3\|$

$$\begin{aligned} T_3 =& \frac{\beta\eta}{m} \sum_{i \in C_1} \mathbf{u}_i^{(t)} \\ =& \frac{\beta\eta}{m} \left(\sum_{i \in C_1 \cap C_1^*} \mathbf{u}_i^{(t)} + \sum_{i \in C_1 \cap \overline{C_1^*}} \mathbf{u}_i^{(t)} \right) \\ =& \underbrace{\frac{\eta\beta}{m} \sum_{i \in C_1 \cap C_1^*} \mathbf{u}_i^{(t)}}_{T_{31}} + \underbrace{\frac{\beta\eta}{m} \sum_{i \in C_1 \cap \overline{C_1^*}} \mathbf{u}_i^{(t)}}_{T_{32}}. \end{aligned} \tag{A18}$$

Because $\mathbf{u}_i^{(t)} = \beta \mathbf{u}_i^{(t-1)} + g_i^{(t-1)}$ and $\mathbf{u}_i^{(0)} = \mathbf{0}$, we can recursively use it t times, yielding

$$\mathbf{u}_i^{(t)} = \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} g_i^{(\tau)}, \forall t \geq 1 \tag{A19}$$

Therefore, substituting (A19) into (A18), yields

$$T_{31} = \frac{\eta\beta}{m} \sum_{i \in C_1 \cap C_1^*} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} g_i^{(\tau)}, \tag{A20}$$

$$T_{32} = \frac{\eta\beta}{m} \sum_{i \in C_1 \cap \overline{C_1^*}} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} g_i^{(\tau)}. \tag{A21}$$

We first analyze T_{31},

$$\begin{aligned} T_{31} =& \frac{\eta\beta}{m} \sum_{i\in C_1\cap C_1^*} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} g_i^{(\tau)} \\ =& \frac{\eta\beta}{m} \sum_{i\in C_1\cap C_1^*} \left(\sum_{\tau=0}^{t-1} \beta^{t-1-\tau} g_i^{(\tau)} - \nabla\phi^1(\chi_1) \right) + \frac{\eta\beta}{m} |C_1\cap C_1^*| \nabla\phi^1(\chi_1). \end{aligned} \tag{A22}$$

According to Assumption 3, we have

$$\begin{aligned} & \mathbb{E}\left[\left\| \sum_{i\in C_1\cap C_1^*} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} g_i^{(\tau)} - \nabla\phi^1(\chi_1) \right\|^2 \right] \\ \leq & \mathbb{E}\left[\left\| \frac{1}{1-\beta} \sum_{i\in C_1\cap C_1^*} \sum_{\tau=0}^{t-1} g_i^{(\tau)} - \nabla\phi^1(\chi_1) \right\|^2 \right] \\ \leq & \frac{1}{(1-\beta)^2} \mathbb{E}\left[\left\| \sum_{i\in C_1\cap C_1^*} \sum_{\tau=0}^{t-1} g_i^{(\tau)} - \nabla\phi^1(\chi_1) \right\|^2 \right] \\ \leq & \frac{1}{(1-\beta)^2} |C_1\cap C_1^*| \frac{v^2}{n'} t, \end{aligned} \tag{A23}$$

which implies

$$\mathbb{E}\left[\left\| \sum_{i\in C_1\cap C_1^*} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} g_i^{(\tau)} - \nabla\phi^1(\chi_1) \right\| \right] \leq \frac{v}{(1-\beta)} \sqrt{|C_1\cap C_1^*| \frac{t}{n'}}. \tag{A24}$$

Therefore, by Markov's inequality, for any $\lambda_3 \in (0,1)$, with a probability of at least $1-\lambda_3$,

$$\left\| \sum_{i\in C_1\cap C_1^*} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} g_i^{(\tau)} - \nabla\phi^1(\chi_1) \right\| \leq \frac{v}{(1-\beta)\lambda_3} \sqrt{|C_1\cap C_1^*| \frac{t}{n'}}. \tag{A25}$$

As can be seen from the above, $|C_1\cap C_1^*| \geq \frac{1}{4} l_1 m$ with a probability of at least $1-2\exp(-clm)$, and choosing $\eta = \frac{1}{L}$. Then according to Assumption 1, we obtain

$$\|\nabla\phi^1(\chi_1) - \nabla\phi^1(\chi_1^*)\| \leq L\|\chi_1 - \chi_1^*\| \leq L. \tag{A26}$$

Combining (A25), (A26) with (A22), and acquiring with a probability of at least $1-\lambda_3-2\exp(-clm)$,

$$\begin{aligned} \|T_{31}\| \leq & \frac{\beta v\sqrt{lt}}{2\lambda_3(1-\beta)L\sqrt{mn'}} + \frac{l}{4} \\ = & c_3 \frac{\beta v\sqrt{lt}}{\lambda_3(1-\beta)L\sqrt{mn'}} + c_4 l. \end{aligned} \tag{A27}$$

Analyzing T_{32}, we define

$$\begin{aligned} T_{3p} =& \sum_{i\in C_1\cap C_p^*} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau} g_i^{(\tau)} \\ =& \sum_{i\in C_1\cap C_p^*} \left(\sum_{\tau=0}^{t-1} \beta^{t-1-\tau} g_i^{(\tau)} - \nabla\phi_p(\chi_1) \right) + |C_1\cap C_p^*| \nabla\phi_p(\chi_1), \end{aligned} \tag{A28}$$

and we can know $T_{32} = \frac{\beta\eta}{m} \sum_{p=2}^{P} T_{3p}$. According to Assumption 3, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\sum_{i\in C_1\cap C_p^*}\sum_{\tau=0}^{t-1}\beta^{t-1-\tau}g_i^{(\tau)}-\nabla\phi_p(\chi_1)\right\|^2\right]\\
\leq&\mathbb{E}\left[\left\|\frac{1}{1-\beta}\sum_{i\in C_1\cap C_p^*}\sum_{\tau=0}^{t-1}g_i^{(\tau)}-\nabla\phi_p(\chi_1)\right\|^2\right]\\
\leq&\frac{1}{(1-\beta)^2}\mathbb{E}\left[\left\|\sum_{i\in C_1\cap C_p^*}\sum_{\tau=0}^{t-1}g_i^{(\tau)}-\nabla\phi_p(\chi_1)\right\|^2\right]\\
\leq&\frac{1}{(1-\beta)^2}|C_1\cap C_p^*|\frac{v^2}{n'}t,
\end{aligned}\tag{A29}$$

which implies

$$\mathbb{E}\left[\left\|\sum_{i\in C_1\cap C_p^*}\sum_{\tau=0}^{t-1}\beta^{t-1-\tau}g_i^{(\tau)}-\nabla\phi_p(\chi_1)\right\|\right]\leq\frac{v}{(1-\beta)}\sqrt{|C_1\cap C_p^*|\frac{t}{n'}},\tag{A30}$$

and for any $\lambda_4\in(0,1)$, by Markov's inequality, acquiring a probability of at least $1-\lambda_4$,

$$\left\|\sum_{i\in C_1\cap C_p^*}\sum_{\tau=0}^{t-1}\beta^{t-1-\tau}g_i^{(\tau)}-\nabla\phi_p(\chi_1)\right\|\leq\frac{v}{(1-\beta)\lambda_4}\sqrt{|C_1\cap C_p^*|\frac{t}{n'}}.\tag{A31}$$

Conclusively, using union bound and (A28), we can conclude with a probability of at least $1-P\lambda_4$,

$$\|T_{32}\|\leq\frac{3L\beta\eta}{m}|C_1\cap\overline{C_1^*}|+\frac{\beta\eta v\sqrt{P}}{m(1-\beta)\lambda_4}\sqrt{|C_1\cap\overline{C_1^*}|\frac{t}{n'}}.\tag{A32}$$

We can substitute (A16) into (A32) with a probability of at least $1-\lambda_2-P\lambda_4$, and choosing $\eta=\frac{1}{L}$, we obtain

$$\begin{aligned}
\|T_{32}\|&\leq c_1\frac{3\beta\sigma^2}{\mu^2\lambda_2\Delta^4n'}+\frac{\beta\sigma v\sqrt{P}}{n'(1-\beta)\mu\Delta^2\lambda_4L}\sqrt{\frac{c_1t}{m\lambda_2}}\\
&=c_5\frac{\beta\sigma^2}{\mu^2\lambda_2\Delta^4n'}+c_6\frac{\beta\sigma v\sqrt{Pt}}{\lambda_4\mu L\Delta^2n'(1-\beta)\sqrt{m\lambda_2}}.
\end{aligned}\tag{A33}$$

Combining (A27) and (A33), and with a probability of at least $1-\lambda_2-\lambda_3-P\lambda_4-2\exp(-clm)$,

$$\|T_3\|\leq\ c_3\frac{\beta v\sqrt{lt}}{\lambda_3(1-\beta)L\sqrt{mn'}}+c_4l+c_5\frac{\beta\sigma^2}{\mu^2\lambda_2\Delta^4n'}+\ c_6\frac{\beta\sigma v\sqrt{Pt}}{\lambda_4\mu L\Delta^2n'(1-\beta)\sqrt{m\lambda_2}}.\tag{A34}$$

Combining (A8), (A17) and (A34) with a probability of at least $1-\lambda_0-P\lambda_1-2\lambda_2-\lambda_3-P\lambda_4-4\exp(-clm)$,

$$\begin{aligned}
\|\chi_1^{(t+1)}-\chi_1^*\|\leq&(1-\frac{l\mu}{8L})\|\chi_1^{(t)}-\chi_1^*\|+\frac{2v}{\lambda_0L\sqrt{lmn'}}\\
&+c_1\frac{\sigma^2}{\mu^2\lambda_2\Delta^4n'}+c_2\frac{v\sigma\sqrt{P}}{\lambda_1\mu L\Delta^2n'\sqrt{m\lambda_2}}\\
&+c_3\frac{\beta v\sqrt{lt}}{\lambda_3(1-\beta)L\sqrt{mn'}}+c_4l+c_5\frac{\beta\sigma^2}{\mu^2\lambda_2\Delta^4n'}\\
&+c_6\frac{\beta\sigma v\sqrt{Pt}}{\lambda_4\mu L\Delta^2n'(1-\beta)\sqrt{m\lambda_2}}.
\end{aligned}\tag{A35}$$

We let $\lambda_5 = \lambda_0 + P\lambda_1 + 2\lambda_2 + \lambda_3 + P\lambda_4 + 4\exp(-clm)$, and

$$\begin{aligned}\rho_0 =& \frac{2v}{\lambda_0 L\sqrt{lmn'}} + c_1\frac{\sigma^2}{\mu^2\lambda_2\Delta^4 n'} + c_2\frac{v\sigma\sqrt{P}}{\lambda_1\mu L\Delta^2 n'\sqrt{m\lambda_2}} \\ &+ c_3\frac{\beta v\sqrt{lt}}{\lambda_3(1-\beta)L\sqrt{mn'}} + c_4 l + c_5\frac{\beta\sigma^2}{\mu^2\lambda_2\Delta^4 n'} \\ &+ c_6\frac{\beta\sigma v\sqrt{Pt}}{\lambda_4\mu L\Delta^2 n'(1-\beta)\sqrt{m\lambda_2}},\end{aligned} \tag{A36}$$

Therefore, by union bound, for any $\lambda_5 \in (0,1)$ and all $p \in [P]$, we can obtain that with a probability of at least $1 - P\lambda_5$,

$$\|\chi_p^{(t+1)} - \chi_p^*\| \leq (1 - \frac{l\mu}{8L})\|\chi_p^{(t)} - \chi_p^*\| + \rho_0. \tag{A37}$$

Appendix B. Proof of Theorem 1

We formally analyze the convergence of our entire algorithm. By choosing

$$\rho_0 \leq \frac{l}{32}(\frac{\mu}{L})^{\frac{3}{2}}\Delta, \tag{A38}$$

to satisfy (A36) and $\|\chi_p^{(t+1)} - \chi_p^*\| \leq \frac{1}{4}\sqrt{\frac{\mu}{L}}\Delta$. Moreover, we iterate T times over (A37) and obtain

$$\|\chi_p^{(T)} - \chi_p^*\| \leq (1 - \frac{l\mu}{8L})^T\|\chi_p^{(0)} - \chi_p^*\| + \frac{8L}{l\mu}\rho_0. \tag{A39}$$

When we choose $T = \frac{8L}{l\mu}\log(\frac{l\mu\Delta}{32\rho_0 L})$, we have

$$\begin{aligned}(1 - \frac{l\mu}{8L})^T\|\chi_p^{(0)} - \chi_p^*\| \leq& \frac{1}{4}\sqrt{\frac{\mu}{L}}\Delta(1 - \frac{l\mu}{8L})^T \\ =& \frac{1}{4}\sqrt{\frac{\mu}{L}}\Delta(1 - \frac{l\mu}{8L})^{\frac{8L}{l\mu}\log(\frac{l\mu\Delta}{32\rho_0 L})} \\ =& \frac{1}{4}\sqrt{\frac{\mu}{L}}\Delta\exp^{-\log(\frac{l\mu\Delta}{32\rho_0 L})} \\ =& \frac{8\rho_0}{l}\sqrt{\frac{L}{\mu}},\end{aligned} \tag{A40}$$

which means

$$\|\chi_p^{(T)} - \chi_p^*\| \leq \frac{8\rho_0}{l}\sqrt{\frac{L}{\mu}} + \frac{8L}{\mu l}\rho_0 \leq \frac{16L}{\mu l}\rho_0. \tag{A41}$$

Accordingly, we can obtain the final rate of convergence

$$\rho = \frac{16L}{\mu l}\rho_0. \tag{A42}$$

References

1. Neglia, G.; Calbi, G.; Towsley, D.; Vardoyan, G. The Role of Network Topology for Distributed Machine Learning. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 2350–2358. [CrossRef]
2. Cheng, P.; Ma, C.; Ding, M.; Hu, Y.; Lin, Z.; Li, Y.; Vucetic, B. Localized Small Cell Caching: A Machine Learning Approach Based on Rating Data. *IEEE Trans. Commun.* **2019**, *67*, 1663–1676. [CrossRef]
3. Xie, J.; Zheng, Y.; Du, R.; Xiong, W.; Cao, Y.; Ma, Z.; Cao, D.; Guo, J. Deep Learning-Based Computer Vision for Surveillance in ITS: Evaluation of State-of-the-Art Methods. *IEEE Trans. Veh. Technol.* **2021**, *70*, 3027–3042. [CrossRef]

4. Chen, H.; Liu, Z.; Kang, X.; Nishide, S.; Ren, F. Investigating voice features for Speech emotion recognition based on four kinds of machine learning methods. In Proceedings of the 6th IEEE International Conference on Cloud Computing and Intelligence Systems, CCIS 2019, Singapore, 19–21 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 195–199. [CrossRef]
5. Manjunath, K.E. *Multilingual Phone Recognition in Indian Languages—Studies in Speech Signal Processing, Natural Language Understanding, and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2022. [CrossRef]
6. Zheng, C.; Liu, S.; Huang, Y.; Zhang, W.; Yang, L. Unsupervised Recurrent Federated Learning for Edge Popularity Prediction in Privacy-Preserving Mobile-Edge Computing Networks. *IEEE Internet Things J.* **2022**, *9*, 24328–24345. [CrossRef]
7. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv* **2016**, arXiv:1610.05492.
8. 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]
9. Caldas, S.; Wu, P.; Li, T.; Konečný, J.; McMahan, H.B.; Smith, V.; Talwalkar, A. LEAF: A Benchmark for Federated Settings. *arXiv* **2018**, arXiv:1812.01097.
10. Xu, C.; Liu, S.; Yang, Z.; Huang, Y.; Wong, K.K. Learning Rate Optimization for Federated Learning Exploiting Over-the-Air Computation. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3742–3756. [CrossRef]
11. Wu, H.; Wang, P. Node Selection Toward Faster Convergence for Federated Learning on Non-IID Data. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 3099–3111. [CrossRef]
12. Li, X.; Huang, K.; Yang, W.; Wang, S.; Zhang, Z. On the Convergence of FedAvg on Non-IID Data. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
13. Cho, Y.J.; Wang, J.; Chiruvolu, T.; Joshi, G. Personalized Federated Learning for Heterogeneous Clients with Clustered Knowledge Transfer. *arXiv* **2021**, arXiv:2109.08119.
14. Shu, J.; Yang, T.; Liao, X.; Chen, F.; Xiao, Y.; Yang, K.; Jia, X. Clustered Federated Multitask Learning on Non-IID Data with Enhanced Privacy. *IEEE Internet Things J.* **2023**, *10*, 3453–3467. [CrossRef]
15. Cho, Y.J.; Wang, J.; Chirvolu, T.; Joshi, G. Communication-Efficient and Model-Heterogeneous Personalized Federated Learning via Clustered Knowledge Transfer. *IEEE J. Sel. Top. Signal Process.* **2023**, *17*, 234–247. [CrossRef]
16. Xie, B.; Dong, X.; Wang, C. An Improved K -Means Clustering Intrusion Detection Algorithm for Wireless Networks Based on Federated Learning. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 9322368:1–9322368:15. [CrossRef]
17. Dennis, D.K.; Li, T.; Smith, V. Heterogeneity for the Win: One-Shot Federated Clustering. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Virtual Event, 18–24 July 2021; Meila, M., Zhang, T., Eds.; Proceedings of Machine Learning Research (PMLR): London, UK, 2021; Volume 139, pp. 2611–2620.
18. Zeng, D.; Hu, X.; Liu, S.; Yu, Y.; Wang, Q.; Xu, Z. Stochastic Clustered Federated Learning. *arXiv* **2023**, arXiv:2303.00897.
19. Awasthi, P.; Sheffet, O. Improved Spectral-Norm Bounds for Clustering. In Proceedings of the Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques—15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, 15–17 August 2012; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7408, pp. 37–49. [CrossRef]
20. Sattler, F.; Müller, K.; Samek, W. Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 3710–3722. [CrossRef] [PubMed]
21. Xie, M.; Long, G.; Shen, T.; Zhou, T.; Wang, X.; Jiang, J.; Zhang, C. Multi-Center Federated Learning. *arXiv* **2021**, arXiv:2005.01026.
22. Ma, Z.; Zhao, M.; Cai, X.; Jia, Z. Fast-convergent federated learning with class-weighted aggregation. *J. Syst. Archit.* **2021**, *117*, 102125. [CrossRef]
23. Ghosh, A.; Chung, J.; Yin, D.; Ramchandran, K. An Efficient Framework for Clustered Federated Learning. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, Virtual, 6–12 December 2020.
24. Ma, J.; Long, G.; Zhou, T.; Jiang, J.; Zhang, C. On the Convergence of Clustered Federated Learning. *arXiv* **2022**, arXiv:2202.06187.
25. Liu, W.; Chen, L.; Chen, Y.; Zhang, W. Accelerating Federated Learning via Momentum Gradient Descent. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 1754–1766. [CrossRef]
26. Luo, X.; Qin, W.; Dong, A.; Sedraoui, K.; Zhou, M. Efficient and High-quality Recommendations via Momentum-incorporated Parallel Stochastic Gradient Descent-Based Learning. *IEEE CAA J. Autom. Sin.* **2021**, *8*, 402–411. [CrossRef]
27. Raghuwanshi, S.K.; Pateriya, R.K. Accelerated Singular Value Decomposition (ASVD) using momentum based Gradient Descent Optimization. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *33*, 447–452. [CrossRef]
28. Ramezani-Kebrya, A.; Khisti, A.; Liang, B. On the Generalization of Stochastic Gradient Descent with Momentum. *arXiv* **2018**, arXiv:1809.04564.
29. Duchi, J.C.; Hazan, E.; Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
30. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
31. McMahan, H.B.; Streeter, M.J. Adaptive Bound Optimization for Online Convex Optimization. In Proceedings of the COLT 2010—The 23rd Conference on Learning Theory, Haifa, Israel, 27–29 June 2010; Omnipress: Madison, WI, USA, 2010; pp. 244–256.
32. Reddi, S.J.; Kale, S.; Kumar, S. On the Convergence of Adam and Beyond. *arXiv* **2019**, arXiv:1904.09237.

33. Luo, L.; Xiong, Y.; Liu, Y.; Sun, X. Adaptive Gradient Methods with Dynamic Bound of Learning Rate. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
34. Reddi, S.J.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečný, J.; Kumar, S.; McMahan, H.B. Adaptive Federated Optimization. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Vienna, Austria, 4 May 2021.
35. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
36. Sutskever, I.; Martens, J.; Dahl, G.E.; Hinton, G.E. On the importance of initialization and momentum in deep learning. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013; Volume 28, pp. 1139–1147.
37. Yan, Y.; Yang, T.; Li, Z.; Lin, Q.; Yang, Y. A Unified Analysis of Stochastic Momentum Methods for Deep Learning. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, 13–19 July 2018; pp. 2955–2961. [CrossRef]
38. Polyak, B. Some methods of speeding up the convergence of iteration methods. *USSR Comput. Math. Math. Phys.* **1964**, *4*, 1–17. [CrossRef]
39. Gong, B.; Xing, T.; Liu, Z.; Wang, J.; Liu, X. Adaptive Clustered Federated Learning for Heterogeneous Data in Edge Computing. *Mob. Netw. Appl.* **2022**, *27*, 1520–1530. [CrossRef]
40. Gauthier, F.; Gogineni, V.C.; Werner, S.; Huang, Y.; Kuh, A. Clustered Graph Federated Personalized Learning. In Proceedings of the 56th Asilomar Conference on Signals, Systems, and Computers, ACSSC 2022, Pacific Grove, CA, USA, 31 October–2 November 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 744–748. [CrossRef]
41. Pye, S.K.; Yu, H. Personalised Federated Learning: A Combinational Approach. *arXiv* **2021**, arXiv:2108.09618.
42. Mansour, Y.; Mohri, M.; Ro, J.; Suresh, A.T. Three Approaches for Personalization with Applications to Federated Learning. *arXiv* **2020**, arXiv:2002.10619.
43. Nafea, M.S.; Shin, E.; Yener, A. Proportional Fair Clustered Federated Learning. In Proceedings of the IEEE International Symposium on Information Theory, ISIT 2022, Espoo, Finland, 26 June–1 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 2022–2027. [CrossRef]
44. Nesterov, Y.E. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Sov. Math. Dokl.* **1983**, *27*, 372–376.
45. Ghadimi, E.; Feyzmahdavian, H.R.; Johansson, M. Global convergence of the Heavy-ball method for convex optimization. In Proceedings of the 2015 European Control Conference (ECC), Linz, Austria, 15–17 July 2015; IEEE: Piscataway, NJ, USA, 2015.
46. Yang, T.; Lin, Q.; Li, Z. Unified Convergence Analysis of Stochastic Momentum Methods for Convex and Non-convex Optimization. *arXiv* **2016**, arXiv:1604.03257.
47. Alahmari, F. A Comparison of Resampling Techniques for Medical Data Using Machine Learning. *J. Inf. Knowl. Manag.* **2020**, *19*, 2040016:1–2040016:13. [CrossRef]
48. Krizhevsky, A.; Hinton, G. Learning multiple layers of features from tiny images. *Handb. Syst. Autoimmune Dis.* **2009**, 1–60.
49. Deng, L. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
50. Wu, R.; Scaglione, A.; Wai, H.; Karakoç, N.; Hreinsson, K.; Ma, W. Federated Block Coordinate Descent Scheme for Learning Global and Personalized Models. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, Virtual, 2–9 February 2021; AAAI Press: Washington, DC, USA, 2021; pp. 10355–10362.
51. Agbinya, J.I. 11 Convolutional Neural Networks. In *Applied Data Analytics—Principles and Applications*; CRC Press: Boca Raton, FL, USA, 2019; pp. 185–204.
52. Wanchen, L. Analysis on the Weight initialization Problem in Fully-connected Multi-layer Perceptron Neural Network. In Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, China, 23–25 October 2020; pp. 150–153. [CrossRef]
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; IEEE Computer Society: Piscataway, NJ, USA, 2016; pp. 770–778. [CrossRef]