FedWCM: Unleashing the Potential of Momentum-based Federated Learning in Long-Tailed Scenarios

Tianle Li*
2200271015@email.szu.edu.cn
Shenzhen University
Shenzhen, China

Yongzhi Huang*
huangyongzhi@email.szu.edu.cn
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China

Linshan Jiang†
linshan@nus.edu.sg
National University of Singapore
Singapore, Singapore

Qipeng Xie
qxieaf@connect.ust.hk
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China

Chang Liu
liuc0063@e.ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Wenfeng Du
duwf@szu.edu.cn
Shenzhen University
Shenzhen, China

Lu Wang‡
wanglu@szu.edu.cn
Shenzhen University
Shenzhen, China

Kaishun Wu
wuks@hkust-gz.edu.cn
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China

Abstract

Federated Learning (FL) enables decentralized model training while preserving data privacy. Despite its benefits, FL faces challenges with non-identically distributed (non-IID) data, especially in long-tailed scenarios with imbalanced class samples. Momentum-based FL methods, often used to accelerate FL convergence, struggle with these distributions, resulting in biased models and making FL hard to converge. To understand this challenge, we conduct extensive investigations into this phenomenon, accompanied by a layer-wise analysis of neural network behavior. Based on these insights, we propose FedWCM, a method that dynamically adjusts momentum using global and per-round data to correct directional biases introduced by long-tailed distributions. Extensive experiments show that FedWCM resolves non-convergence issues and outperforms existing methods, enhancing FL's efficiency and effectiveness in handling client heterogeneity and data imbalance.

CCS Concepts

• **Computing methodologies** → **Machine learning**; *Federated learning*; Distributed algorithms; Imbalanced learning.

*These authors contributed equally to this work.
†Co-corresponding author. For academic inquiries, please contact this author.
‡Main corresponding author.

ICPP '25, San Diego, CA, USA

ACM ISBN 979-8-4007-2074-1/25/09
https://doi.org/10.1145/3754598.3754657

ACM Reference Format:
Tianle Li, Yongzhi Huang, Linshan Jiang, Qipeng Xie, Chang Liu, Wenfeng Du, Lu Wang, and Kaishun Wu. 2025. FedWCM: Unleashing the Potential of Momentum-based Federated Learning in Long-Tailed Scenarios. In *Proceedings of the 54th International Conference on Parallel Processing (ICPP '25), September 8–11, 2025, San Diego, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3754598.3754657

1 Introduction

Federated Learning (FL) [26] enables collaborative model training across multiple parties without centralizing data, thus ensuring privacy by sharing only model updates with a central server, which aggregates and redistributes a global model. A challenge in these environments is the non-independently and identically distributed (non-IID) data across parties [48], which can slow convergence and reduce performance [19]. Solutions include data augmentation [13, 17], personalized federated learning [9], and clustering [31]. Momentum-based methods—applied at the server [30, 36], client [18, 43], or both [38]—help mitigate non-IID issues by speeding convergence through historical gradient accumulation, offering a simple and efficient solution without additional computation.

However, in real-world situations, lots of non-IID data may meet a long-tailed distribution [35], where the global class distribution is imbalanced: head classes contain abundant samples, whereas tail classes have relatively few, leading to a bias in client models towards the head classes [33]. For example, in some IoT applications, such as smart homes and healthcare monitoring systems, common activities such as sitting and walking dominate the data. At the same time, critical events like falls or specific medical conditions are rare. As the complexity of addressing long-tailed non-IID data will be significantly increased, it remains an open problem that is far from being resolved [3].

So far, there are very limited studies focusing on non-IID data collocated with long-tailed data distribution to tackle the dual challenges [46]. BalanceFL [35] corrects local training through a local update scheme, forcing the local model to behave as if it were trained on a uniform distribution. The Fed-GraB method [42] addresses global long-tailed distribution by employing a self-adjusting gradient balancer and a prior analyzer. CReFF [32] alleviates the bias by retraining classifiers on federated features. CLIP2FL [34] leverages the strengths of the CLIP model to enhance feature representation. However, on the one hand, these methods primarily mitigate issues arising from long-tail distributions without any special design for the convergence speed; on the other hand, these approaches often require disruptive modifications to the methods themselves, making it challenging to integrate with other potential techniques to accelerate the convergence process.

To tackle these challenges, we aim to unleash the potential of the momentum-based approach to improve the learning performance on long-tailed non-IID data distribution. We attempt to adopt a naive momentum-based approach [4, 43] as a start point, and we find that they perform terribly in various settings, where the convergence details are shown in the motivation. It brings us a question: *can we design novel plug-in modules that can fit momentum-based FL algorithms to deal with the long-tailed non-IID data challenges while keeping their strength?*

We delve into this question by first analyzing the convergence difficulties caused by long-tailed distributions, identifying the core issue as momentum-induced global direction distortion. To tackle this issue, we propose an improved momentum-based federated learning approach called FedWCM that introduces a novel momentum adjustment mechanism. This mechanism consists of two key adaptive strategies: first, we adjust the aggregation method of momentum by utilizing global insights to refine how momentum is collected and integrated across clients; second, we modify the degree to which momentum is applied, dynamically tailoring it based on comprehensive global and local data assessments. By implementing these strategies, we ensure that the momentum mechanism effectively mitigates the negative impacts of long-tailed data distributions. This dual approach preserves momentum's acceleration benefits while addressing the inherent challenges.

Our main contributions are summarized as follows:

- To the best of our knowledge, we are the first to identify the convergence challenges faced by momentum-based federated learning under long-tailed non-IID data distributions. Through intuitive analysis, we have demonstrated that this issue arises since the momentum induced causes the global aggregation direction to be skewed by the bias introduced by long-tailed data, thereby hindering model convergence.
- Inspired by the above insights, we propose FedWCM, in which global information is incorporated to adaptively adjust the previously fixed momentum value and momentum aggregation weights on a per-round basis, resulting in a balanced global momentum and enabling it to exert an appropriate influence in the next round based on its situation. With this novel design, FedWCM can leverage the advantages of momentum while avoiding the non-convergence issues that arise in long-tailed scenarios.
- We perform the convergence analysis and conduct extensive experiments to demonstrate the effectiveness of FedWCM. Our theoretical proof indicates that FedWCM has the same convergence rate as FedCM [43]. Furthermore, our experiments result shows that FedWCM exceeds the performance of state-of-the-art algorithms on various datasets and outperforms some potential long-tail enhancement methods when integrated with FedCM.

2 Related work

2.1 Momentum-based Federated Learning

To address client drift in heterogeneous federated learning, researchers have proposed various momentum-based methods. SCAFFOLD [19] introduces control variates to correct client update biases, Mime [18] combines momentum SGD with variance reduction techniques, FedDyn [1] balances global and local objectives through dynamic regularization, and SlowMo [39] incorporates momentum updates on the server side. These methods mitigate client drift and improve model convergence through different momentum strategies.

2.2 Federated Long-tailed Learning

Federated long-tailed learning [22] aims to address global class imbalance issues in federated environments. Some studies explore federated long-tailed learning solutions from a meta-learning perspective [29], aiming to improve model adaptability to new tasks. Research has also focused on client selection and aggregation strategy improvements [12, 47] to better handle long-tailed data. Additionally, some researchers [41] have attempted to address the challenge of model personalization, aiming to improve the accuracy of local models in long-tailed environments. While these methods have made progress in mitigating class imbalance issues, they primarily focus on addressing problems arising from long-tailed data, with less consideration given to client heterogeneity.

There are limited studies working on non-IID data collocated with long-tailed data distribution. Model decoupling methods like FedGraB [42] characterize global long-tailed distributions under privacy constraints and adjust local learning strategies. Feature enhancement methods have been explored, with CReFF [32] retraining classifiers on federated features, and CLIP2FL [34] enhancing client feature representation through knowledge distillation and prototype contrastive learning. Some studies explore federated long-tailed learning solutions from a meta-learning perspective [29], aiming to improve model adaptability to new tasks. While these methods have made progress in addressing this problem, some practical data partitioning scenarios are not considered, and there is no specific design to improve algorithm efficiency.

Several approaches that addresses the local long-tailed distribution in centralized machine learning may have the potential to be integrated with momentum-based FL approaches for long-tailed non-IID data distribution. Researchers have proposed various methods to tackle local long-tailed challenge. Re-balancing strategies such as Focal loss [24], PriorCELoss [16] and LDAM loss [2] adjust prediction probabilities or introduce label-distribution-aware margin losses to improve tail class performance. However, experiments

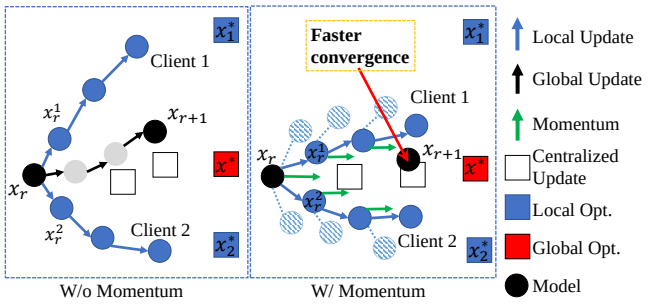

Figure 1: Client drift due to heterogeneity without momentum and alleviation with momentum.

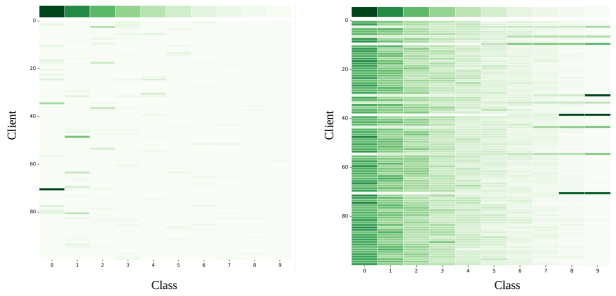

Figure 2: Client data partition on CIFAR-10: FedGrab v.s. ours, when $\beta = 0.1$ and $IF = 0.1$.

will show that the naive integration does not have sufficient effect to address this issue.

3 Preliminaries

3.1 Federated Learning with Momentum

Federated Learning (FL) allows multiple clients to train a global model collaboratively without sharing their local data. Mathematically, FL optimizes:

$$\min_{w} \left[F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \right] \tag{1}$$

where $F_k(w)$ is the local loss function of client k, n_k is the number of data samples at client k, and w represents the global model parameters.

FL faces challenges due to data heterogeneity, leading to client drift, where inconsistent model updates slow or prevent convergence. To address this, momentum-based approaches [43] [4] has been proposed during local training:

$$\mathbf{v}_k^r = \alpha \mathbf{g}_k^r + (1-\alpha)\Delta_r \tag{2}$$

where $\mathbf{v}_k^r$ is the momentum of client k at round r, α is the momentum value, and Δ_r is the global momentum obtained from the average of client gradients in the previous round.

Figure 1 illustrates how momentum mitigates client drift and accelerates convergence. In detail, the momentum leverages the aggregated gradients from the previous round to obtain global directional information, aligning each client's update direction during local training. This alignment of client gradient updates effectively mitigates client drift.

3.2 Long-Tailed and non-IID Data Distribution

Long-tailed data distributions are practical in real-world scenarios and characterized by a significant imbalance between the most and least frequent classes. We define the imbalance factor IF [2, 33] as: $IF = \frac{n_1}{n_C}$ where n_1 and n_C are the global sample counts of the most and least frequent classes, respectively. When IF is smaller, the tales of the global data distributions is longer.

To introduce non-IID data distribution on the clients, we use Dirichlet allocation [26]: $p_{k,c} \sim \mathrm{Dir}(\beta)$ where $p_{k,c}$ is the proportion of samples for class c allocated to client k, and β controls the degree of heterogeneity. Note that the smaller β denotes higher skew.

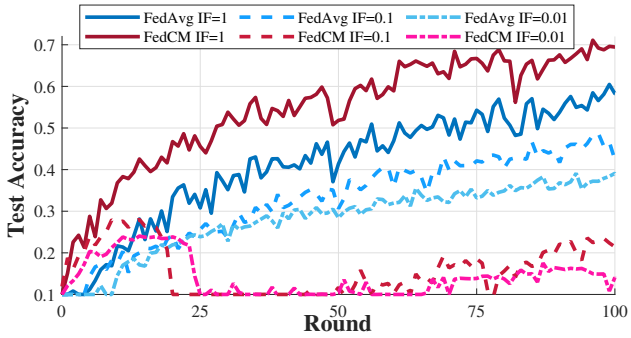

Figure 3: Test accuracy over communication rounds on CIFAR-10 with $\beta = 0.1$ and various settings of IF.

While existing works in LT-problem [3] and FedGrab [42] leverage a similar long-tailed distributions, they often result in inconsistent client data quantities. That is to say, their data partition naturally incur high skew on data quantities. However, in some practical IoT scenarios [35], the number of the data samples are similar among different clients. Thus, in this paper, we follow the partition strategy [35] with β and IF to designate the training data to clients, as shown in the right part in Figure 2. The discussion of the data partition shown in the left figure is in Appendix A [23][1].

4 Motivation

Momentum-based methods like FedCM [43] introduce global momentum to guide local training and reduce client drift, showing promise for faster convergence and higher accuracy compared to FedAvg under certain conditions (e.g., Dirichlet distribution with $\beta = 0.1$ and $IF = 1$, as shown in Figure 3). However, while momentum can align aggregated gradients with the global objective in class-balanced scenarios, it becomes a double-edged sword in long-tailed data distributions, where it amplifies majority-class gradients, exacerbates data imbalance, and risks non-convergence. This issue is evident in the red and pink lines of Figure 3, which show FedCM failing to converge under $IF = 0.1$ and $IF = 0.01$.

[1]Appendix and full technical proofs are available at https://li-tian-le.github.io/FedWCM-Supplement/

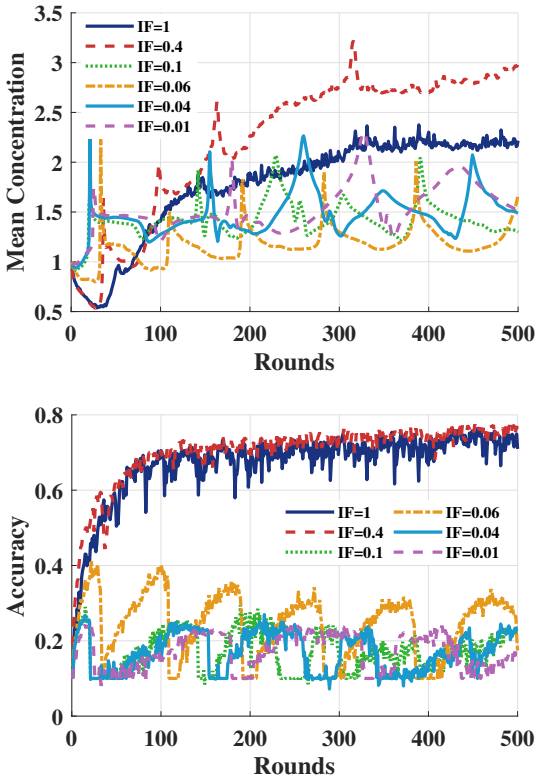

Figure 4: Both figures show results across six different imbalance factor (*IF*) settings. Top: Average neuron concentration change in FedCM. Bottom: Test accuracy across the same six *IF* settings.

As shown in Figure 4, the top plot illustrates the mean activation concentration under different imbalance factor (IF) settings. Under balanced conditions (e.g., $IF = 1$), the neuron concentration increases steadily, aligning with the theory of neural collapse [10, 20, 44]. However, under long-tailed distributions, we observe abrupt spikes in neuron concentration at certain critical points. Notably, as the imbalance increases (i.e., as IF decreases), these spikes occur more frequently and violently.

Through detailed analysis in Appendix B [23], we infer that these sudden increases are caused by the momentum mechanism [37], which leads to dominant neurons corresponding to majority classes occupying the representational space of others. These spikes are not random fluctuations but structured transitions in the optimization dynamics, where momentum amplifies class-specific gradients in a biased way. This results in a phenomenon known as **Minority Collapse** [10], where the network overfits to majority classes, causing the representation of minority classes to degrade sharply.

In this situation, the model's capacity to accurately classify minority classes diminishes significantly, leading to a steep drop in test accuracy. This effect is illustrated in the bottom plot of Figure 4, showing a synchronous decline in accuracy as the minority representations collapse. These insights underline the importance of mitigating momentum-induced instability, which motivates the design of our proposed method.

5 Methodology

We propose FedWCM to tackle non-convergence in momentum-based federated learning with long-tailed distributions. FedWCM dynamically adjusts client weights and momentum, reducing the dominance of majority-class clients and enhancing the impact of minority-class clients while modulating momentum to maintain stable convergence.

5.1 Global Information Gathering

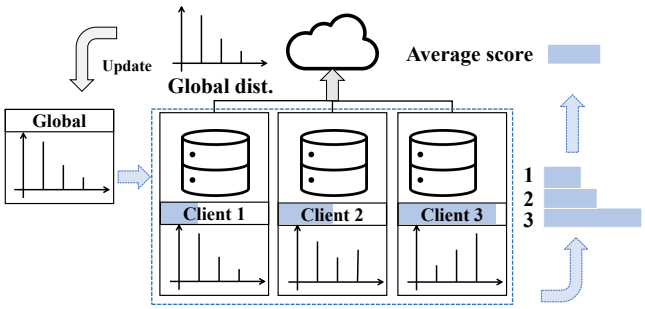

Figure 5: Illustration of global information gathering.

Clients need global knowledge of the data distribution to adjust their local updates. As illustrated in Figure 5, we compute and distribute the global data distribution D_g to all clients. Each client computes a score based on the deviation of its local distribution from the global target distribution. The score for client k is:

$$s_k = \frac{\sum_{c=1}^{C} |\hat{p}_c - p_c| \cdot n_{k,c}}{\sum_{c=1}^{C} n_{k,c}} \tag{3}$$

where $\hat{p}_c$ is the proportion of class c in the global target distribution, p_c is the proportion of class c in the global distribution, and $n_{k,c}$ is the number of samples of class c for client k. By default, the global target distribution is assumed to be uniform, but users can adjust it based on the prior distribution relevant to their specific application scenarios.

A higher score indicates that the client has more globally scarce data. This scoring helps clients adjust their updates with a global perspective, mitigating the impact of data heterogeneity.

5.2 Parameter Computation

Next, client weights and adaptive momentum values are computed. As illustrated in Figure 6, client weights are computed using a modified Softmax function:

$$w_k^r = \frac{\exp(s_k^r/T)}{\sum_{j \in \mathcal{P}_r} \exp(s_j^r/T)} \tag{4}$$

where T is a temperature parameter that works inversely with the imbalance: as the data imbalance increases, the temperature decreases, which increases the differences between client weights. This allows clients with more representative data to have more influence during the training process. Conversely, when the global data imbalance is low, the temperature increases, resulting in more uniform client weights. In practice, T is computed based on the

discrepancy between the target (usually uniform) distribution and the actual global data distribution, scaled appropriately by the number of classes to maintain consistent sensitivity across different datasets. This mechanism helps balance the influence of each client on the model during training.

The adaptive momentum value α_{r+1} is calculated as:

$$\alpha_{r+1} = 0.1 + 0.9 \cdot (1 - e^{-\|\mathbf{T}/\mathbf{K}\|_1}) \cdot q_r \tag{5}$$

where q_r is the ratio between the average client score sampled in the current round $\hat{s}^r$ and the overall average score $\hat{s}$. The base momentum α_0 is set to 0.1. The ratio q_r reflects the degree to which minority classes are represented in the current round's sampled clients relative to the entire client population. When the current round's average client score is relatively high, indicating better minority class representation, the momentum is increased to leverage more informative gradient updates. This ensures that momentum is dynamically adjusted based on the data imbalance across rounds.

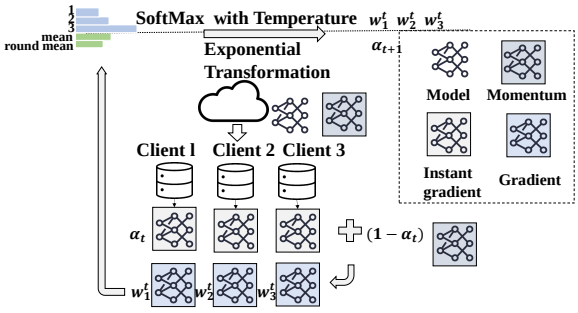

Figure 6: Parameter computation and application.

5.3 Parameter Application in Training

The computed parameters are applied to local model updates during each round. Clients update their models using the formula:

$$x^r_{b+1,k} = x^r_{b,k} - \eta_l(\alpha_r \nabla F(x^r_{b,k}; D_{b,k}) + (1 - \alpha_r)\Delta_r) \tag{6}$$

where α_r controls the impact of the current round's gradient. This dynamic adjustment helps balance the effects of majority-class and minority-class clients, improving model robustness across rounds.

5.4 Overall FedWCM Algorithm

In summary, we obtain global data distribution to dynamically set parameters for each round's momentum aggregation and application process, enabling momentum to alleviate, rather than exacerbate, the negative impacts of data imbalance. Through integrated mechanisms, our method effectively addresses the challenges posed by long-tail data distributions, ensuring improved federated learning outcomes. The pseudo algorithm is shown in Algorithm 1.

5.5 Privacy Discussion

Note that our approach requires access to the global data distribution. While some works [6, 11, 25, 47] emphasize protecting local data distributions, others [7, 8, 27] tolerate the potential leakage of local data distributions for algorithmic design purposes. Nevertheless, our method is compatible with existing privacy-preserving techniques, such as Homomorphic Encryption (HE), to safeguard local data distributions. Following the approach in BatchCrypt [45], the HE-based protocol proceeds as follows:

- **Key Generation:** A subset of randomly selected clients generates public-private key pairs and distributes the public keys to the other clients.
- **Encryption and Upload:** Each client encrypts their local class distribution using the received public key and uploads the encrypted data to the server.
- **Aggregation:** The server aggregates the encrypted class distributions using homomorphic operations, without decrypting them.
- **Decryption and Reconstruction:** The aggregated ciphertext is sent to the corresponding key generator, who decrypts it to obtain the global class distribution and uploads it to the server.

This protocol assumes a semi-honest server (i.e., honest-but-curious) and does not require any trusted third-party entities. We further explore this issue in Appendix C [23].

Algorithm 1 FedWCM Algorithm

Require: initial model x_0, global momentum Δ_0, $\alpha_0 = 0.1$, learning rates η_l, η_g, number of rounds R, local iterations B
Compute $\{s_k\}$ with D_g using Equation (3)
for $r = 0$ to $R - 1$ **do**
  Sample subset $\mathcal{P}_r$ of clients
  for Each client $k \in \mathcal{P}_r$ **do**
    $x^r_{0,k} = x_r$
    for $b = 0$ to $B - 1$ **do**
      Compute $g^r_{b,k} = \nabla f_k(x^r_{b,k}, D_{b,k})$
      $v^r_{b,k} = \alpha_r g^r_{b,k} + (1 - \alpha_r)\Delta_r$
      $x^r_{b+1,k} = x^r_{b,k} - \eta_l v^r_{b,k}$
    end for
    $\Delta^r_k = x^r_{B,k} - x_r$
  end for
  Compute w^r_k using Equation (4)
  Compute α_{r+1} using Equation (5)
  $\Delta_{r+1} = \frac{1}{\eta_l B} \sum_{k \in \mathcal{P}_r} w^r_k \Delta^r_k$
  $x_{r+1} = x_r - \eta_g \Delta_{r+1}$
end for

5.6 Method Generalization

The original FedWCM algorithm assumes that clients possess roughly equal data volumes. However, in real-world scenarios, this assumption often fails due to data quantity heterogeneity. To address this, we extend FedWCM to better handle high quantity skew, resulting in a generalized version termed **FedWCM-X**, as shown in Algorithm 2.

FedWCM-X introduces two key modifications to enhance robustness under non-uniform data distribution.

First, we incorporate an additional weighting factor based on each client's data volume. Specifically, for a client k, if its original

Algorithm 2 FedWCM-X Algorithm

Require: initial model x_0, global momentum Δ_0, $\alpha_0 = 0.1$, learning rates η_l, η_g, number of rounds R, local iterations B, standard iterations $\hat{B}$
Compute $\{s_k\}$ with D_g using Equation (3)
for $r = 0$ to $R-1$ **do**
  Sample subset $\mathcal{P}_r$ of clients
  for Each client $k \in \mathcal{P}_r$ **do**
    $x^r_{0,k} = x_r$
    $\eta'_l = \eta_l \cdot \frac{\hat{B}}{B_k}$
    for $b = 0$ to $B_k - 1$ **do**
      Compute $g^r_{b,k} = \nabla f_k(x^r_{b,k}, D_{b,k})$
      $v^r_{b,k} = \alpha_r g^r_{b,k} + (1-\alpha_r)\Delta_r$
      $x^r_{b+1,k} = x^r_{b,k} - \eta'_l v^r_{b,k}$
    end for
    $\Delta^r_k = x^r_{B_k,k} - x_r$
  end for
  Compute w^r_k using Equation (4)
  $w'^r_k = w^r_k \cdot \frac{n_k}{\sum_j n_j}$
  Compute α_{r+1} using Equation (5)
  $\Delta_{r+1} = \frac{1}{\eta_l \hat{B}} \sum_{k \in \mathcal{P}_r} w'^r_k \Delta^r_k$
  $x_{r+1} = x_r - \eta_g \Delta_{r+1}$
end for

update weight is w_k, the adjusted weight is defined as:

$$w'_k = w_k \cdot \frac{n_k}{\sum_j n_j}$$

where n_k denotes the number of local data samples held by client k. This adjustment ensures that clients with more data exert proportionally more influence during model aggregation.

Second, we normalize the local learning rate according to the number of mini-batches each client executes. Since clients with more data may perform more local updates, we rescale the learning rate η_l to prevent instability caused by excessively large gradient steps. The adjusted learning rate is given by:

$$\eta'_l = \eta_l \cdot \frac{\hat{B}}{B_k}$$

where B_k is the number of local batches for client k, and $\hat{B}$ is a reference batch count corresponding to a balanced data split across all clients.

Together, these modifications enable FedWCM-X to retain the benefits of momentum-based correction while adapting to imbalanced client participation, improving convergence stability and model performance in practical federated environments. The effectiveness of FedWCM-X under non-uniform data quantity settings is demonstrated through experiments presented in Appendix A [23].

6 Convergence Analysis

In this section, we prove that the convergence rate of FedWCM is the same as FedAvg-M [4], e.g., $\sqrt{\frac{L\Delta\sigma^2}{NKR}} + \frac{L\Delta}{R}$.

Assumption 1 (Standard Smoothness). *Each local objective function f_i is L-smooth, i.e., for any $x, y \in \mathbb{R}^d$ and $1 \le i \le N$, we have*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x - y\|. \tag{7}$$

Assumption 2 (Stochastic Gradients). *There exists $\sigma \ge 0$ such that for any $x \in \mathbb{R}^d$ and $1 \le i \le N$, we have*

$$\mathbb{E}_{\xi_i}[\nabla F(x; \xi_i)] = \nabla f_i(x), \tag{8}$$

and

$$\mathbb{E}_{\xi_i}[\|\nabla F(x; \xi_i) - \nabla f_i(x)\|^2] \le \sigma^2, \tag{9}$$

where $\xi_i \sim \mathcal{D}_i$ are independent and identically distributed.

Under the assumptions stated above, we provide the convergence result for FedWCM with adaptive β and weighted aggregation. The following theorem summarizes the key convergence guarantee.

Theorem 6.1 (Convergence of FedWCM). *Let $f(x)$ be the global objective function. Under Assumptions 1 and 2, with $g_0 = 0$, $\beta \le \sqrt{\frac{NKL\Delta}{\sigma^2 R}}$ for any constant $c \in (0, 1]$, $\gamma = \min\left(\frac{1}{24L}, \frac{\beta}{6L}\right)$, and $\eta KL \lesssim \min\left(1, \frac{1}{\beta\gamma LR}, \left(\frac{L\Delta}{G_0\beta^3 R}\right)^{1/2}, \frac{1}{(\beta N)^{1/2}}, \frac{1}{(\beta^3 NK)^{1/4}}\right)$, FedWCM achieves the following convergence rate:*

$$\frac{1}{R}\sum_{r=0}^{R-1} \mathbb{E}[\|\nabla f(x_r)\|^2] \lesssim \sqrt{\frac{L\Delta\sigma^2}{NKR}} + \frac{L\Delta}{R}, \tag{10}$$

where $\Delta = f(x_0) - \min_x f(x)$, and $\lesssim$ absorbs constant numerical factors. Here, $G_0 := \frac{1}{N}\sum_{i=1}^N \|\nabla f_i(x_0)\|^2$ represents the average squared norm of the client gradients at the initial point x_0.

The proof of Theorem 6.1 extends the standard convergence analysis of FedAvg-M [4] with two key modifications: an adaptive β and weighted aggregation. The parameter β is dynamically adjusted during training, constrained within $[0.1, 1)$, and satisfies $\beta \le \sqrt{\frac{NKL\Delta}{\sigma^2 R}}$, ensuring stability across different data distributions. Additionally, the adaptive aggregation weights mitigate the bias term U_r by accounting for discrepancies between local and global data distributions. These changes preserve the original convergence guarantees while improving adaptability. Further details, including bias analysis, are provided in Appendix E [23].

7 Experiments

7.1 Experimental Setup

We mainly conduct experiments on Fashion-MNIST [40], SVHN [28], CIFAR-10 [21], CIFAR-100 [21] and ImageNet [5] datasets. For the data partition, we adopt the partition strategy as shown in the preliminary part. By default, we set $p_{k,c} \sim \mathrm{Dir}(\beta)$, where $\beta = 0.1$. For the imbalanced ratio IF, the default setting is $IF = 0.1$. Note that smaller β denotes worse skews while smaller IF denotes higher class imbalance level.

By default, we use a multilayer perceptron (MLP) architecture for the Fashion-MNIST dataset. For the SVHN and CIFAR-10 datasets, we employ ResNet-18 [15] as the backbone network. For the CIFAR-100 and ImageNet datasets, we use ResNet-34 [15]. We set the batch size to 50, the local learning rate η_l to 0.1, the global learning rate η_g to 1, and the local epoch to 5. By default, the number of clients

Table 1: Performance comparison under $\beta = 0.6$ and $\beta = 0.1$ across different datasets and imbalance factors (IF). We report the mean test accuracy under 3 trials on different random seeds.

		FedAvg		BalanceFL		FedCM		FedCM + Focal Loss		FedCM + Balance Loss		FedCM + Balance Sampler		FedWCM	
Dataset	IF	0.6	0.1	0.6	0.1	0.6	0.1	0.6	0.1	0.6	0.1	0.6	0.1	0.6	0.1
Fashion-MNIST	1	**0.8800**	0.8074	0.8795	**0.8443**	0.8419	0.7604	0.8246	0.6931	0.7451	0.6907	0.8546	0.7821	0.8625	0.8181
	0.5	**0.8688**	0.8079	0.8638	0.8462	0.8601	**0.8544**	0.8363	0.7058	0.7622	0.6737	0.8476	0.7906	0.8659	0.8366
	0.1	0.8450	0.8313	**0.8497**	**0.8475**	0.8211	0.8268	0.8161	0.8065	0.7873	0.8002	0.8245	0.8252	0.8469	0.8328
	0.05	0.8318	0.8408	**0.8520**	**0.8545**	0.3914	0.4975	0.1945	0.1967	0.4335	0.4064	0.5474	0.5273	0.8499	0.8426
	0.01	0.7871	0.7894	**0.8192**	**0.8126**	0.7378	0.7524	0.7188	0.7265	0.8039	0.8011	0.8027	0.8030	0.7882	0.7947
SVHN	1	0.9361	0.8986	**0.9370**	0.9146	0.9246	0.8836	0.9242	0.8749	0.8928	0.8312	0.9310	0.8911	0.9355	**0.9276**
	0.5	0.9251	0.9271	0.9261	0.9243	0.7137	0.6981	0.6068	0.5961	0.5594	0.5870	0.7085	0.6761	**0.9324**	**0.9284**
	0.1	0.8681	0.8741	0.8979	0.8976	0.1594	0.0670	0.1959	0.1959	0.0762	0.1322	0.1959	0.1976	**0.9057**	**0.9024**
	0.05	0.8594	0.8647	0.8675	0.8709	0.3100	0.0723	0.0670	0.1107	0.0969	0.0909	0.1802	0.0932	**0.8759**	**0.8836**
	0.01	0.7884	0.7803	0.7901	0.7954	0.0670	0.0670	0.0670	0.0751	0.0760	0.0759	0.1736	0.2427	**0.7998**	**0.8408**
CIFAR-10	1	0.7906	0.6881	0.7629	0.6813	0.8126	0.7092	0.8040	0.6937	0.7931	0.7169	0.8065	0.7198	**0.8242**	**0.7337**
	0.5	0.7535	0.7183	0.7539	0.7429	0.6793	0.6686	0.6565	0.6319	0.6877	0.6924	0.6968	0.6590	**0.7926**	**0.7968**
	0.1	0.6232	0.6775	0.6380	0.6541	0.2175	0.2393	0.1311	0.3095	0.1864	0.3016	0.2871	0.3994	**0.6905**	**0.7207**
	0.05	0.5715	0.5642	0.5652	0.5535	0.2274	0.2358	0.2005	0.1413	0.2680	0.2525	0.1427	0.1315	**0.6006**	**0.6132**
	0.01	0.4567	0.4600	0.4731	0.4616	0.1865	0.2312	0.1687	0.2023	0.2087	0.2405	0.1249	0.1584	**0.4983**	**0.5012**
CIFAR-100	1	0.4297	0.3731	0.3691	0.3232	0.4129	0.2400	0.3990	0.2357	0.3630	0.2089	0.3599	0.2339	**0.4545**	**0.3858**
	0.5	0.3545	0.3882	0.3203	0.3639	0.2996	0.4200	0.3058	0.3853	0.2694	0.3722	0.2835	0.3790	**0.4195**	**0.4202**
	0.1	0.2839	0.2744	0.2440	0.2407	0.2948	0.3135	0.3014	0.3166	0.2952	0.3156	0.2952	0.2955	**0.3150**	**0.3235**
	0.05	0.2155	0.2300	0.2070	0.2157	0.1130	0.2695	0.0100	0.2806	0.1000	0.2786	0.0930	0.2721	**0.2573**	**0.2832**
	0.01	0.1663	0.1885	0.1565	0.1609	0.0116	0.1035	0.0109	0.1027	0.0100	0.1286	0.0100	0.0723	**0.1985**	**0.2005**
ImageNet	1	0.2760	0.2290	0.2292	0.1947	0.2479	0.1408	0.2438	0.1222	0.2082	0.1024	0.2134	0.1155	**0.3094**	**0.2462**
	0.5	0.2154	0.2140	0.1628	0.2124	0.1045	0.0392	0.0923	0.0695	0.0928	0.0544	0.1154	0.1067	**0.2598**	**0.2198**
	0.1	0.1631	0.1535	0.1124	0.1161	0.1796	0.1738	0.1864	0.1763	0.1796	0.1788	0.1528	0.1521	**0.1923**	**0.1874**
	0.05	0.1458	0.1355	0.0915	0.0998	0.0052	0.1597	0.1355	0.1448	0.1471	0.1576	0.1130	0.1542	**0.1626**	**0.1660**
	0.01	0.0882	0.1123	0.0627	0.0612	0.0050	0.1137	0.0063	0.1354	0.0050	0.1209	0.0052	0.1217	**0.0974**	**0.1383**

is set to 100 with a participation rate of 0.1 and 500 communication rounds for each experiment. For CIFAR-100 and ImageNet, the default number of clients is set to 40, and we conduct 300 communication rounds. Additional experimental settings can be found in the corresponding figures/tables. Note that all experiments were implemented in PyTorch and conducted on a workstation equipped with four NVIDIA GeForce RTX 3090 GPUs.

7.2 Overall Accuracy Evaluation

In this experiment, we compare FedWCM against several representative baselines to evaluate its effectiveness in handling long-tailed and heterogeneous federated learning scenarios. We consider two main categories of methods: (1) standard and long-tail-specific federated learning methods, including FedAvg [26], BalanceFL [35], and FedGrab [42]; (2) improved variants of FedCM incorporating common imbalance handling techniques such as Focal Loss [24], Balance Loss [16], and Balance Sampler [14]. Experiments are conducted on Fashion-MNIST, SVHN, CIFAR-10, CIFAR-100, and ImageNet, under Dirichlet distributions with $\beta = 0.1$ and $\beta = 0.6$ to simulate varying degrees of non-IID data.

Table 1 summarizes the results on CIFAR-10. Across almost all settings, FedWCM achieves the highest accuracy, demonstrating strong generalization ability and robustness under both mild and severe imbalance. For instance, at $\beta = 0.6$, $IF = 0.1$, FedWCM reaches 0.6905 accuracy, outperforming FedAvg (0.6232), BalanceFL (0.638), and FedGrab (0.326). The performance gap becomes more pronounced as imbalance increases.

While FedGrab performs competitively under moderate heterogeneity (e.g., $IF = 1$, $IF = 0.5$), it degrades significantly in highly imbalanced or heterogeneous settings. Particularly under $\beta = 0.1$, its accuracy drops sharply — for example, only 32.60% at $IF = 0.1$, compared to FedAvg's 67.75% and FedWCM's 72.07%. This highlights FedGrab's sensitivity to severe non-IID distributions and limited robustness.

FedAvg, while not specifically designed for long-tail data, provides a stable baseline across most scenarios, though it generally underperforms compared to more specialized methods. BalanceFL shows slight improvements over FedAvg, but its performance remains inconsistent.

As for FedCM and its variants, these methods generally fail to converge or produce very low accuracy in long-tailed settings. FedCM alone yields only 0.2175 accuracy in some cases. Even with the introduction of Focal Loss, Balance Loss, or Balance Sampler, accuracy remains low (e.g., 0.1311, 0.1864, and 0.2871, respectively), indicating that these improvements are insufficient to resolve convergence issues in the presence of severe imbalance.

It is also worth noting that the effectiveness of FedWCM varies depending on the dataset and model complexity. On Fashion-MNIST, the simpler 3-layer MLP architecture limits the benefits of momentum-based strategies. Moreover, datasets with more classes (e.g., CIFAR-100 and ImageNet) show slightly reduced performance gains for FedWCM due to diluted long-tail effects across many classes.

In summary, FedWCM consistently outperforms existing methods, including FedGrab and FedAvg, particularly under harsh conditions. Its adaptive momentum and weight correction mechanism ensure stable and accurate learning in federated environments with long-tailed, heterogeneous data.

7.3 Efficiency Evaluation of FedWCM

Convergence experiments. As illustrated in Figure 7, under the settings of $\beta = 0.6$ and $IF = 0.1$, FedWCM in the orange line demonstrates a very rapid convergence in test accuracy during the early stages, quickly achieving a high accuracy level. FedWCM's

Table 2: Performance comparison on CIFAR-10 under $\beta = 0.6$ and $\beta = 0.1$ with varying imbalance factors (IF). Results are averaged over 3 runs with different seeds.

		FedAvg		FedGrab		FedWCM	
Dataset	IF	0.6	0.1	0.6	0.1	0.6	0.1
CIFAR-10	1	0.7906	0.6881	0.7950	0.6813	**0.8242**	**0.7337**
	0.5	0.7535	0.7183	0.7810	0.6560	**0.7926**	**0.7968**
	0.1	0.6232	0.6775	0.6880	0.3260	**0.6905**	**0.7207**
	0.05	0.5715	0.5642	0.5000	0.1870	**0.6006**	**0.6132**
	0.01	0.4567	0.4600	0.3140	0.1350	**0.4983**	**0.5012**

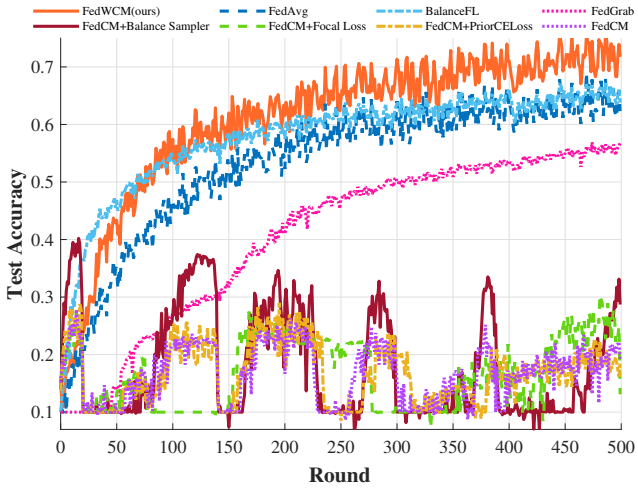

Figure 7: Test accuracy on various methods ($\beta = 0.6$, $IF = 0.1$).

performance continues to improve, maintaining a high accuracy range, and eventually stabilizing around 400 iterations, reaching a test accuracy of 75%. Compared to other methods such as FedAvg and FedGrab, FedWCM consistently maintains a leading position throughout most of the iterations. Conversely, different variants of FedCM exhibit characteristics of non-convergence.

Besides, compared to most algorithms, FedWCM demonstrates a rapid increase in test accuracy during the initial stages of iteration, surpassing the 60% accuracy threshold at around 120 iterations. In comparison, BalanceFL approaches this level at around 200 iterations, while FedAvg requires approximately 300 iterations to reach the same accuracy. In contrast, FedGrab fails to achieve the 60% accuracy threshold even after more than 500 iterations. Additionally, the convergence speed of FedWCM is significantly faster than that of other converging algorithms.

Per-label accuracy pattern. As shown in Figure 8, with $IF = 0.1$ and $\beta = 0.6$, FedWCM achieves high accuracy on minor labels, notably outperforming FedAvg and FedCM on labels 6, 7, 8, and 9. Figure 2 shows the global label distribution, where label 1 is the most frequent and label 9 the least. This highlights FedWCM's strength in handling long-tailed distributions. In contrast, FedCM's accuracy drops sharply with decreasing label frequency, nearing zero for label 9.

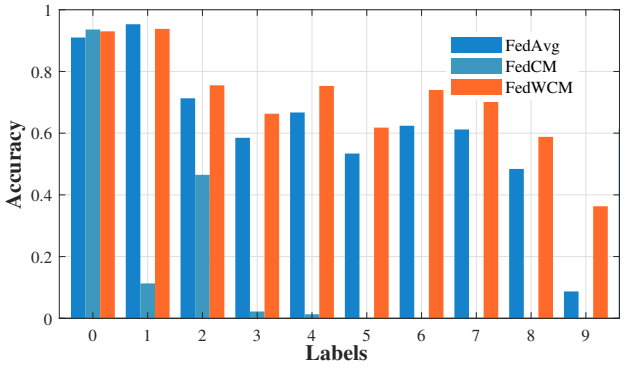

Figure 8: Per-label accuracy comparison of various methods when $\beta = 0.6$ and $IF = 0.1$.

Table 3: Comparison under different client smapling rates.

Sampling Rate	FedAvg	FedCM	FedWCM
5%	0.6865	0.3130	**0.7127**
10%	0.6232	0.1918	**0.6905**
20%	0.6450	0.3006	**0.7164**
40%	0.6418	0.2268	**0.6933**
80%	0.6441	0.1000	**0.6980**

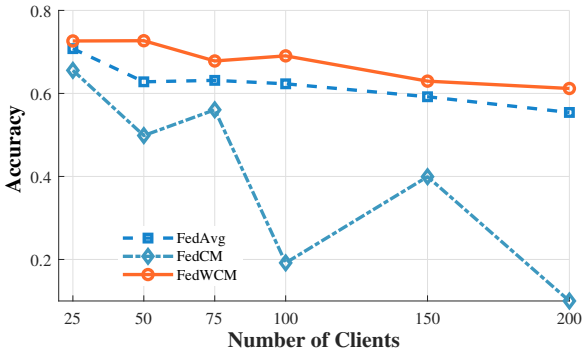

Figure 9: Test accuracy w.r.t the number of clients.

7.4 Scalability Analysis

Client Participation Rate. FedWCM maintains relatively high accuracy across all levels of client participation rates shown in Table 3, particularly at lower participation rates (i.e, 5% and 10%), where its accuracy significantly surpasses that of FedAvg and FedCM. Although the accuracy in FedWCM gradually decreases as the participation rate increases (i.e., from 0.7127 at 5% to 0.6025 at 80%), the decline is more gradual compared to FedAvg and FedCM.

Total Client Number. As shown in Figure 9, as the number of clients increases, the amount of data allocated to each client decreases, exacerbating data imbalance under the same IF, which leads to performance degradation for all three algorithms. Among

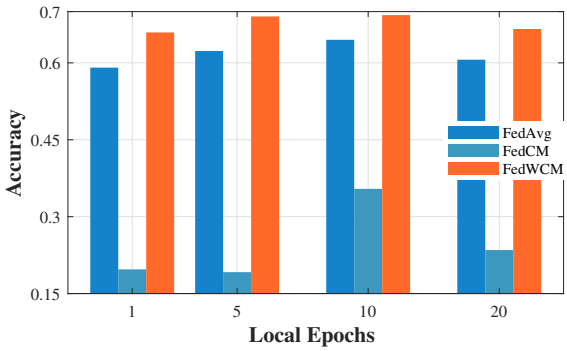

Figure 10: Test accuracy w.r.t local epochs.

them, FedWCM exhibits the slowest decline and maintains a relatively high accuracy (around 0.7). In contrast, FedAvg shows a significant performance drop when the number of clients reaches 50, while FedCM suffers from severe performance fluctuations as the number of clients increases, often resulting in non-convergence.

7.5 Ablation Study

Local Epochs. Figure 10 compares the accuracy of FedAvg, FedCM, and FedWCM across different numbers of local epochs (1, 5, 10, 20 epochs). The results show that FedWCM consistently outperforms the other algorithms across all local round settings, with its accuracy significantly improving as the number of local epochs increases. This performance advantage is particularly evident in mid-to-high round settings. In contrast, FedAvg demonstrates relatively stable performance but consistently falls short of FedWCM, while FedCM displays considerable variability, with accuracy significantly lower than FedWCM across the settings.

Various settings of β and IF. Table 4 presents the performance comparison of FedAvg, FedCM, and FedWCM when $\beta = 0.1$ and $\beta = 0.6$. The FedWCM algorithm demonstrates outstanding performance across all configurations, especially when addressing different Dirichlet parameters and varying information factors, even when other algorithms do not converge. FedWCM shows low sensitivity to the β parameter, maintaining high-performance levels even under more dispersed data distributions. Furthermore, although all algorithms generally exhibit a performance decline as the information factor decreases, FedWCM experiences a relatively minor drop. Particularly at shallow IF values (i.e., $IF = 0.01$), FedWCM remains effective in adapting to sparse data environments, showing its superior adaptability.

7.6 Supplementary Experiments

We provide several supplementary experiments [23] to further validate the effectiveness and robustness of our method. Appendix A presents results of FedWCM-X, a generalized extension of FedWCM designed for scenarios with unequal data volumes among clients, demonstrating its stable performance under such imbalance. Appendix B investigates neuron concentration patterns across FedAvg, FedCM, and FedWCM under varying data distributions, aiming to explore the root causes of non-convergence in long-tailed settings.

Table 4: Comparison of various approaches under different settings.

$\beta = 0.1$	**IF**	**1**	**0.4**	**0.1**	**0.06**	**0.04**	**0.01**
FedAvg		0.6859	0.7059	0.6228	0.6295	0.5358	0.4838
FedCM		0.7179	0.7394	0.2346	0.2077	0.2206	0.2283
FedWCM		**0.7337**	**0.7735**	**0.6629**	**0.6538**	**0.5972**	**0.5078**
$\beta = 0.6$	**IF**	**1**	**0.4**	**0.1**	**0.06**	**0.04**	**0.01**
FedAvg		0.7912	0.7294	0.6232	0.5801	0.5543	0.4637
FedCM		0.8104	0.7363	0.1918	0.2616	0.1894	0.2399
FedWCM		**0.8426**	**0.7969**	**0.6905**	**0.6216**	**0.6042**	**0.5164**

Appendix C examines the resource overhead introduced when integrating FedWCM with homomorphic encryption, showing the approach remains feasible in privacy-sensitive environments. Appendix D further supplements the performance of momentum-based methods against other heterogeneous FL baselines.

8 Conclusion

In this work, we design a novel momentum-based federated learning algorithm called FedWCM to address the convergence challenges in long-tailed non-IID data distributions. By dynamically adjusting momentum aggregation and application, FedWCM effectively mitigates the negative impacts of skewed data distributions, resulting in improved convergence and performance across various datasets. Our extensive experiments demonstrate FedWCM's superiority over state-of-the-art algorithms, making it a robust solution for federated learning in complex and imbalanced data scenarios while leverages the advantages of momentum-based approach.

Acknowledgments

The research was supported in part by the China NSFC Grant (No. 62372307, No. U2001207, No. 62472366), Guangdong NSF (No. 2024A1515011691), Shenzhen Science and Technology Program (No. RCYX20231211090129039), Shenzhen Science and Technology Foundation (No. JCYJ20230808105906014, No. ZDSYS20190902092853047), Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No. 2023B1212010007), 111 Center (No. D25008), and the Project of DEGP (No. 2023KCXTD042, No. 2024GCZX003). Lu WANG is the corresponding author.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. 2021. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263* (2021).

[2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems* 32 (2019).

[3] Zihan Chen, Songshang Liu, Hualiang Wang, Howard H Yang, Tony QS Quek, and Zuozhu Liu. 2022. Towards federated long-tailed learning. *arXiv preprint arXiv:2206.14988* (2022).

[4] Ziheng Cheng, Xinmeng Huang, Pengfei Wu, and Kun Yuan. 2023. Momentum benefits non-iid federated learning simply and provably. *arXiv preprint arXiv:2306.16504* (2023).

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 248–255.

[6] Jian-hui Duan, Wenzhong Li, Derun Zou, Ruichen Li, and Sanglu Lu. 2023. Federated learning with data-agnostic distribution fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8074–8083.

[7] Moming Duan, Duo Liu, Xianzhang Chen, Renping Liu, Yujuan Tan, and Liang Liang. 2020. Self-balancing federated learning with global imbalanced data in mobile systems. *IEEE Transactions on Parallel and Distributed Systems* 32, 1 (2020), 59–71.

[8] Moming Duan, Duo Liu, Xianzhang Chen, Yujuan Tan, Jinting Ren, Lei Qiao, and Liang Liang. 2019. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In *2019 IEEE 37th international conference on computer design (ICCD)*. IEEE, 246–254.

[9] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in neural information processing systems* 33 (2020), 3557–3568.

[10] Cong Fang, Hangfeng He, Qi Long, and Weijie J Su. 2021. Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proceedings of the National Academy of Sciences* 118, 43 (2021), e2103091118.

[11] Yanan Fu, Xuefeng Liu, Shaojie Tang, Jianwei Niu, and Zhangmin Huang. 2021. CIC-FL: enabling class imbalance-aware clustered federated learning over shifted distributions. In *Database Systems for Advanced Applications: 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11–14, 2021, Proceedings, Part I 26*. Springer, 37–52.

[12] DaoQu Geng, HanWen He, XingChuan Lan, and Chang Liu. 2022. Bearing fault diagnosis based on improved federated learning algorithm. *Computing* 104, 1 (2022), 1–19.

[13] Jack Goetz and Ambuj Tewari. 2020. Federated learning via synthetic data. *arXiv preprint arXiv:2008.04489* (2020).

[14] Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering* 21, 9 (2009), 1263–1284.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[16] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. 2021. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6626–6636.

[17] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2018. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479* (2018).

[18] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. 2020. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606* (2020).

[19] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*. PMLR, 5132–5143.

[20] Vignesh Kothapalli. 2022. Neural collapse: A review on modelling principles and generalization. *arXiv preprint arXiv:2206.04041* (2022).

[21] A. Krizhevsky and G. Hinton. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto* (2009).

[22] Kan Li, Yang Li, Ji Zhang, Xin Liu, and Zhichao Ma. 2024. Federated deep long-tailed learning: A survey. *Neurocomputing* 595 (2024), 127906.

[23] Tianle Li, Yongzhi Huang, Linshan Jiang, Qipeng Xie, Chang Liu, Wenfeng Du, Lu Wang, and Kaishun Wu. 2025. Supplementary Material for "FedWCM: Unleashing the Potential of Momentum-based Federated Learning in Long-Tailed Scenarios". https://li-tian-le.github.io/FedWCM-Supplement/. Supplementary material for the ICPP 2025 accepted paper.

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[25] Xiang Liu, Liangxi Liu, Feiyang Ye, Yunheng Shen, Xia Li, Linshan Jiang, and Jialin Li. 2023. FedLPA: Personalized One-shot Federated Learning with Layer-Wise Posterior Aggregation. *arXiv preprint arXiv:2310.00339* (2023).

[26] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[27] Naram Mhaisen, Alaa Awad Abdellatif, Amr Mohamed, Aiman Erbad, and Mohsen Guizani. 2021. Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints. *IEEE Transactions on Network Science and Engineering* 9, 1 (2021), 55–66.

[28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, Vol. 2011. Granada, 4.

[29] Pinxin Qian, Yang Lu, and Hanzi Wang. 2023. Long-Tailed Federated Learning Via Aggregated Meta Mapping. In *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2010–2014.

[30] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295* (2020).

[31] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems* 32, 8 (2020), 3710–3722.

[32] Xinyi Shang, Yang Lu, Gang Huang, and Hanzi Wang. 2022. Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features. *arXiv preprint arXiv:2204.13399* (2022).

[33] Zhiyuan Shang et al. 2022. Addressing Class Imbalance in Federated Learning. *arXiv preprint arXiv:2205.12345* (2022).

[34] Jiangming Shi, Shanshan Zheng, Xiangbo Yin, Yang Lu, Yuan Xie, and Yanyun Qu. 2023. Clip-guided federated learning on heterogeneous and long-tailed data. *arXiv preprint arXiv:2312.08648* (2023).

[35] Xian Shuai, Yulin Shen, Siyang Jiang, Zhihe Zhao, Zhenyu Yan, and Guoliang Xing. 2022. BalanceFL: Addressing class imbalance in long-tail federated learning. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 271–284.

[36] Jianhui Sun, Xidong Wu, Heng Huang, and Aidong Zhang. 2024. On the role of server momentum in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 15164–15172.

[37] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. 2020. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in neural information processing systems* 33 (2020), 1513–1524.

[38] Jiajun Wang, Yingchi Mao, Xiaoming He, Tong Zhou, Jun Wu, and Jie Wu. 2023. Accelerating Federated Learning with Two-phase Gradient Adjustment. In *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 810–817.

[39] Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. 2019. Slowmo: Improving communication-efficient distributed sgd with slow momentum. *arXiv preprint arXiv:1910.00643* (2019).

[40] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[41] Zikai Xiao, Zihan Chen, Liyinglan Liu, Yang Feng, Jian Wu, Wanlu Liu, Joey Tianyi Zhou, Howard Hao Yang, and Zuozhu Liu. 2024. FedLoGe: Joint Local and Generic Federated Learning under Long-tailed Data. *arXiv preprint arXiv:2401.08977* (2024).

[42] Zikai Xiao, Zihan Chen, Songshang Liu, Hualiang Wang, Yang Feng, Jin Hao, Joey Tianyi Zhou, Jian Wu, Howard Yang, and Zuozhu Liu. 2024. Fed-grab: Federated long-tailed learning with self-adjusting gradient balancer. *Advances in Neural Information Processing Systems* 36 (2024).

[43] Jing Xu, Sen Wang, Liwei Wang, and Andrew Chi-Chih Yao. 2021. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874* (2021).

[44] Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. 2022. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? *Advances in neural information processing systems* 35 (2022), 37991–38002.

[45] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. 2020. BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. USENIX Association, 493–506. https://www.usenix.org/conference/atc20/presentation/zhang-chengliang

[46] Jing Zhang, Chuanwen Li, Jianzgong Qi, and Jiayuan He. 2023. A survey on class imbalance in federated learning. *arXiv preprint arXiv:2303.11673* (2023).

[47] Shulai Zhang, Zirui Li, Quan Chen, Wenli Zheng, Jingwen Leng, and Minyi Guo. 2021. Dubhe: Towards data unbiasedness with homomorphic encryption in federated learning client selection. In *Proceedings of the 50th International Conference on Parallel Processing*. 1–10.

[48] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. 2021. Federated learning on non-IID data: A survey. *Neurocomputing* 465 (2021), 371–390.