Technical Report for FedWCM

A Analysis of FedWCM under Different Data Partitioning

We opted for a custom data partitioning approach to ensure that the data across clients is roughly consistent. Existing long-tailed heterogeneous datasets often present challenges due to extreme data imbalance. To verify the applicability of our method on other datasets, we implemented FedGrab [7]'s data partitioning and conducted comparative experiments.

A.1 Problem Description

There is currently no universal method for long-tailed heterogeneous partitioning. BalanceFL [15] uses its own long-tailed heterogeneous partitioning, while CLIP [21] and Creff [13] (using the same partitioning), FedGrab, and our approach first generate a long-tailed dataset, followed by Dirichlet partitioning. However, this methodology has a drawback: when partitioning a long-tailed dataset, the sampling from long-tailed data means that even with Dirichlet sampling, originally majority classes are likely to remain majority classes on clients.

CLIP and Creff, along with FedGrab, provide two partitioning methods, both generating Dirichlet distributions for each class and then allocating to clients. Yet, this can lead to some clients having no data (i.e., proportions in all classes are so low that they do not constitute even one data point, especially when the imbalance factor is small). To address this, the former repeatedly samples until the requirement is met, which may indirectly control the degree of long-tailed distribution, while the latter assigns at least one data point to each client.

For our method, the original momentum-based method [2, 18] initially did not address the issue of inconsistent data quantity due to two reasons: 1) Solving heterogeneity issues does not necessarily require addressing quantity heterogeneity [20], as they primarily target distributional heterogeneity. 2) The momentum base method introduces a fixed global momentum in a weighted manner each round. Therefore, when there is a large disparity in data quantity between clients, more data leads to more batches. This results in multiple additions of momentum in that client, negating the intended effect of reducing client variance.

Secondly, our main text introduces a method that weights based on data distribution. If we use a common method for addressing data quantity heterogeneity, weighting by data quantity, it may overlap with our method, preventing an effective analysis of our method's effects.

Lastly, our comparison in the main text is also justified, because if a method can address data quantity heterogeneity, it should also perform well in non-heterogeneous scenarios. Here, we supplement experiments with partitioning methods that increase data quantity disparity, demonstrating the applicability of our method under various data distributions.

Algorithm 1 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 $\mathscr{P}_r$ of clients
  for Each client $k \in \mathscr{P}_r$ 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)
  $\Delta_{r+1} = \frac{1}{\eta_l \hat{B}} \sum_{k \in \mathscr{P}_r} w''^r_k \Delta^r_k$
  $x_{r+1} = x_r - \eta_g \Delta_{r+1}$
end for

A.2 Method Generalization

Our original method assumed that clients have similar amounts of data. To address scenarios with significant disparities in client data quantities, we proposed an improved FedWCM for high quantity skew, naming FedWCM-X, as shown in Algorithm 1. In detail, we outline the two steps required for this extension:

1. Building upon the existing weighting, we additionally weight by data quantity. Specifically, if the current round's weight is w_k, we now multiply it by $\frac{n_k}{\sum_j n_j}$, where n_k represents the data quantity of the k-th client.

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

2. We adjust the learning rate η_l based on the batch numbers corresponding to different data quantities. This involves dividing η_l by the current batch number B_k, and then multiplying by a standard batch number $\hat{B}$, which is the number of batches a client would have if the data were evenly distributed across clients.

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

A.3 Experimental Section

We illustrate the superiority of our method over six other approaches through an accuracy variation graph on this dataset(as shown in Figure 1), using the partitioning strategy based on FedGrab [7].

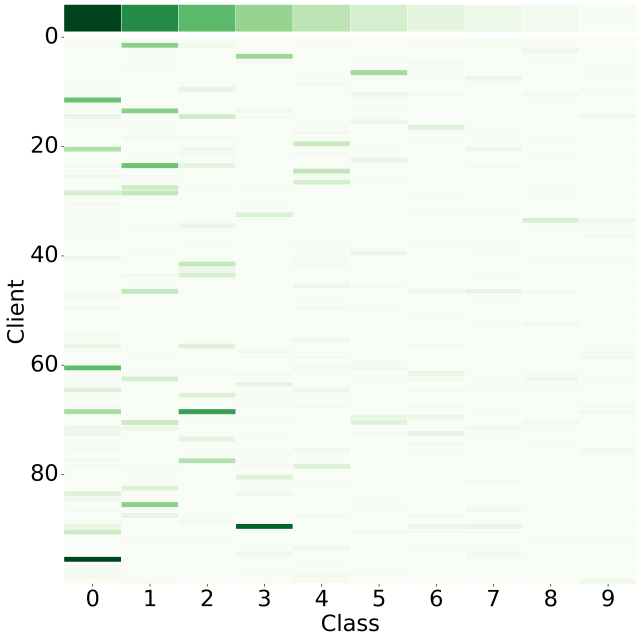

Figure 1: Data distribution under the setting of $\beta = 0.1$, $IF = 0.1$, partitioned according to FedGrab.

The dataset exhibits significant imbalance across clients, with approximately 10% of clients holding over 50% of the total samples, while around 40% of clients possess less than 10% of the samples. Additionally, certain clients predominantly represent a few classes, leading to skewed class distributions. This imbalance not only necessitates robust methods to ensure equitable participation and effective model aggregation across all clients, but also poses significant challenges for the implementation and weighting of momentum. In scenarios with uneven data quantities, momentum methods may be dominated by a few clients with large datasets, potentially impacting the overall model performance.

From Figure 2, we can observe that FedWCM in brown line maintains a high convergence speed in the early stages of training, and its final convergence accuracy is comparable to FedAvg in purple line and BalanceFL in gray line. The lack of its original significant advantage is attributed to the interference caused by weighting based on quantity and inherent weighting, which affects the gradient aggregation effectiveness. On the other hand, the average performance of FedGrab in green line may result from differences in settings such as batch size and number of clients compared to the original experiment. Other potential improvements based on FedCM do not converge.

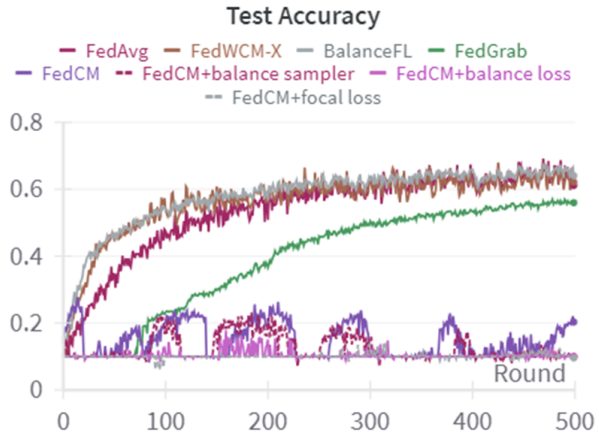

Figure 2: Accuracy comparison of our method against six other methods on the dataset.

Additionally, we compare FedAvg, FedCM, and FedWCM-X under this data partitioning scheme with IF settings of 1, 0.4, 0.1, 0.06, 0.04, and 0.01. The comparison is shown in the following table, focusing on the case where $\beta = 0.1$.

Table 1: Comparison of various approaches under different settings of IFs when $\beta = 0.1$.

	IF	1	0.4	0.1	0.06	0.04	0.01
FedAvg		0.6802	0.7069	0.6219	0.577	0.5502	0.4905
FedCM		0.6696	0.7405	0.2095	0.1527	0.1438	0.1438
FedWCM-X		0.6895	0.7346	0.6236	0.5793	0.5632	0.4911

As shown in Table 1, there are notable performance differences among various approaches under different settings. FedWCM-X consistently outperforms other methods in most scenarios, especially at lower IF values, maintaining the highest accuracy. For instance, at $IF = 0.1$ and $IF = 0.04$, FedWCM-X achieves accuracies of 0.6236 and 0.5632, respectively, significantly surpassing other methods. In contrast, the performance of FedAvg decreases gradually as IF decreases, whereas FedCM performs poorly at low IF values, with a marked drop in accuracy.

B Exploration of Non-Convergence in Momentum-based Methods

In this section, we explore the mechanisms behind the non-convergence of momentum-based methods under long-tailed distributions. Attempts to theoretically prove non-convergence have been challenging due to the complexity of deriving inevitable non-convergence conclusions. In centralized algorithms, as discussed in [17], causal inference has been used to qualitatively analyze the impact of momentum on imbalanced data, suggesting that "bad" effects can be removed while retaining the "good." This provides some theoretical support, indicating the adverse aspects of momentum on imbalanced data.

Our analysis is inspired by the Neural Collapse [5] [11] [19], which provides a rigorous mathematical explanation. Neural Collapse describes a scenario in the terminal phase of training deep neural networks, where classifiers and output features form a special geometric structure called the Simplex Equiangular Tight Frame when training samples are balanced. This structure maximizes the angle between features and classifiers of different classes, minimizing inter-class confusion and explaining the excellent generalization and robustness of deep neural networks.

However, when sample numbers are imbalanced, this geometric structure is disrupted, leading to a new phenomenon called Minority Collapse [5]. Majority classes dominate the loss function, allowing their features and classifiers to span larger angles, while minority classes are compressed, reducing their angles. This phenomenon has been confirmed through mathematical analysis and experiments.

Based on this theory, we observed the concentration of neurons across different layers of the neural network during training. We introduce line charts depicting the average concentration changes over rounds for FedAvg, FedCM, and FedWCM under $\beta = 0.1$, $IF = 1$ on the left, and $\beta = 0.1$, $IF = 0.1$ on the right.

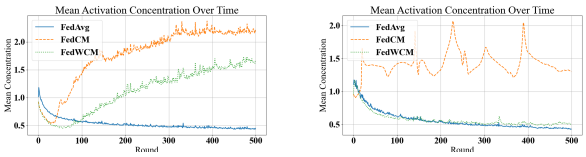

Figure 3: Average neuron concentration over rounds for FedAvg, FedCM, and FedWCM under different settings. Left: $\beta = 0.1$, $IF = 1$. Right: $\beta = 0.1$, $IF = 0.1$.

From the left subfigure with $\beta = 0.1$, $IF = 0$, it can be observed that the average neuron concentration over rounds for FedAvg in blue line gradually decreases, whereas both FedCM in orange line and FedWCM in green line initially decrease and then increase. This may be due to the accumulation of certain neurons' advantages under the influence of momentum. Furthermore, the increase in FedWCM is relatively smooth. The reason for FedWCM's increase is that we adjust the distillation temperature based on the global imbalance; when the global distribution is fairly balanced, we avoid extreme weighting, as experiments have shown that FedCM performs well in non-long-tailed scenarios. From the right plot with $\beta = 0.1$, $IF = 0.1$, it is evident that both FedAvg in blue line and FedWCM in green line exhibit a downward trend in average neuron concentration over rounds, with FedWCM declining faster and more smoothly. In contrast, FedCM in orange line shows periodic large fluctuations.

Next, we introduce three charts showing the detailed concentration changes for each method across layers.

It can be observed that the neuron concentration in all layers for FedAvg shows a downward trend. In contrast, FedCM exhibits periodic large fluctuations in neuron concentration

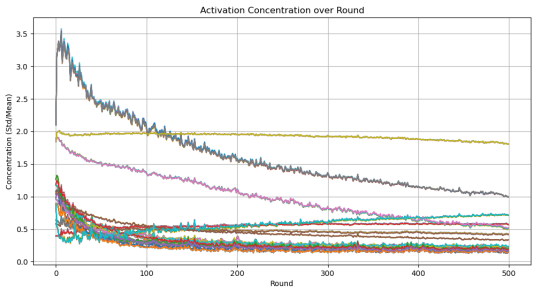

Figure 4: Detailed neuron concentration changes across layers for FedAvg.

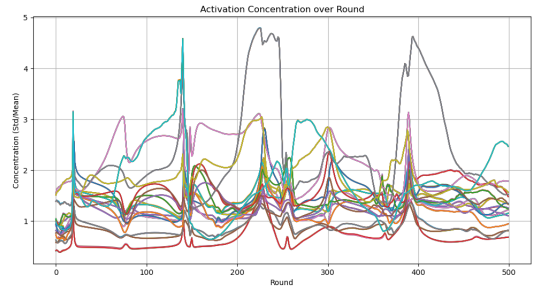

Figure 5: Detailed neuron concentration changes across layers for FedCM.

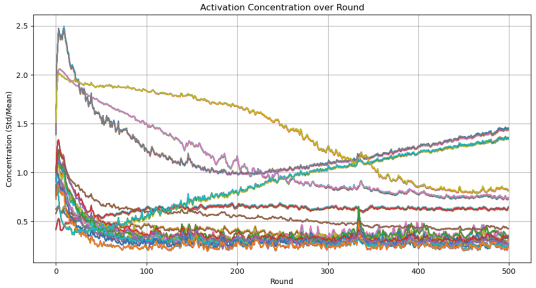

Figure 6: Detailed neuron concentration changes across layers for FedWCM.

across all layers, which might be the underlying reason for its difficulty in converging to a stable point. For FedWCM, the neuron concentration mostly decreases across layers, with some layers experiencing an increase, but overall, it remains very stable.

We then focus on analyzing the neuron concentration in FedCM before and after the critical points under long-tailed scenarios. Here, we introduce a combined image showing the accuracy across five long-tailed scenarios and their average concentration change.

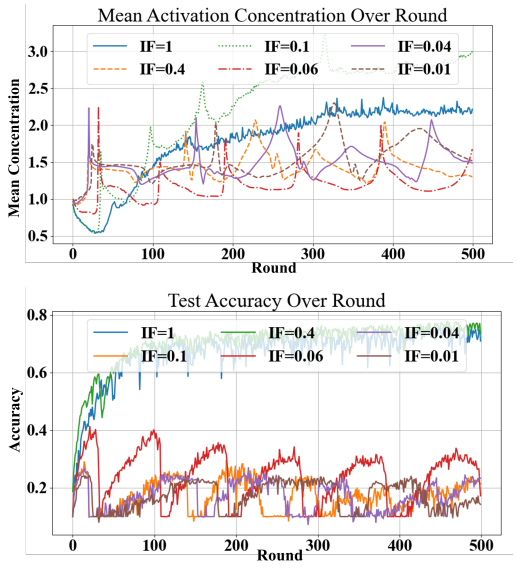

Figure 7: Top: Average neuron concentration change in FedCM. Bottom: Accuracy across five long-tailed scenarios.

By comparing the Average neuron concentration and accuracy change graphs of FedCM under various IF conditions, we observe that as FedCM experiences a precipitous drop in accuracy, its Average neuron concentration also undergoes a sudden change. We believe there is a strong correlation between these two phenomena, possibly due to the occurrence of the minority collapse as discussed in [5].

C Homomorphic Encryption for Data Distribution in FedWCM

To protect the privacy of clients' local class distribution information in FedWCM, we adopt homomorphic encryption (HE), following the protocol used in BatchCrypt [?]. HE enables computations directly on encrypted data, ensuring that operations on ciphertexts yield results consistent with computations on plaintexts [6].

Specifically, an encryption scheme $E(\cdot)$ is said to be additively homomorphic if it satisfies:

$$E(m_1) \oplus E(m_2) = E(m_1 + m_2),$$

and multiplicatively homomorphic if:

$$E(m_1) \odot E(m_2) = E(m_1 \cdot m_2),$$

where $\oplus$ and $\odot$ denote ciphertext-level operations.

In our implementation:

- A randomly selected subset of clients generates public/private key pairs and distributes the public keys to other clients.
- Each participating client encrypts their local class distribution vector using the public key and uploads the ciphertext to the server.
- The server aggregates the ciphertexts and sends the result back to the corresponding key holder for decryption, yielding the global class distribution.

This process assumes a semi-honest server and does not rely on any trusted third party, aligning with the design goals of BatchCrypt.

Since local class distributions are represented as integer vectors, we use the BFV scheme (Brakerski/Fan-Vercauteren) [3], which supports exact arithmetic over integers. Our implementation is based on the TenSEAL library.

To assess communication overhead, we measured the size of both plaintext and ciphertext representations under varying numbers of classes. As shown in Table 2, plaintext size increases linearly with the number of classes, while ciphertext size remains relatively stable at approximately 86KB due to fixed encryption parameters.

Number of Classes	Plaintext (Byte)	Ciphertext (Byte)
10	136	88556
20	216	88554
50	456	88631
100	856	88548

Table 2: Plaintext and ciphertext sizes for different numbers of classes.

Notably, since each client only encrypts their own class distribution, the communication cost is independent of the number of clients. For instance, in a scenario with 100 clients and 10-class distributions, the homomorphic encryption process takes approximately 0.0017 seconds per client, with a total communication size of just 13.05MB—negligible compared to model transmission overhead in a typical federated learning round.

In conclusion, our integration of HE into FedWCM enables secure estimation of global class distributions during early training, helping to mitigate class imbalance in long-tailed scenarios while incurring minimal computation and communication overhead.

D Supplementary Experiments

D.1 Supplementary Experiments for Heterogeneous methods

Figure 8 and Figure 9 compare the performance of FedCM against nine other federated learning methods under heterogeneous data environments. These methods include FedAvg [14], SCAFFOLD [10], FedDyn [1], FedProx [12], FedSAM [22], MoFedSAM [22], FedSpeed [9], FedSMOO [8], and FedLESAM [22]. The experiments were conducted on the CIFAR-10 dataset with a heterogeneity level of $\beta = 0.1$ (non-long-tailed distribution). As shown in the figures, FedCM not only demonstrates significantly faster convergence but also achieves the highest test accuracy (0.71) at 100 communication rounds, outperforming all other methods. This highlights the strong performance of FedCM in heterogeneous data settings.

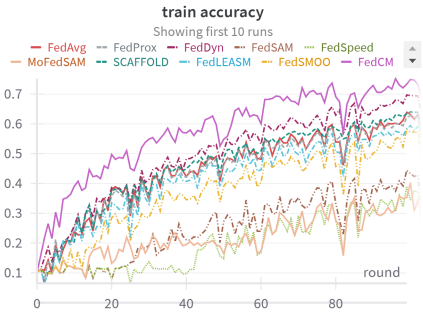

Figure 8: Comparison of Heterogeneous methods for train accuracy.

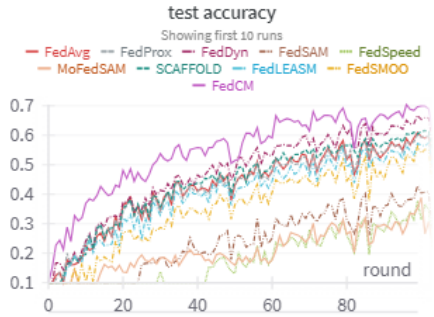

Figure 9: Comparison of Heterogeneous methods for test accuracy.

Using FedAvg [14] as the baseline, several classical methods, such as FedDyn [1], SCAFFOLD [10], and FedProx [12], achieve higher test accuracy. Specifically, FedDyn, with its dynamic regularization strategy, and SCAFFOLD, which uses control variates to correct local updates, effectively mitigate the local drift caused by data heterogeneity. These methods exhibit relatively smooth convergence curves and achieve slightly higher accuracy than FedAvg. However, their final accuracy still falls short of FedCM.

The three SAM-based methods, including FedSAM [22], FedSMOO [16], and FedLESAM [4], focus on improving generalization by flattening the loss landscape. However, as observed in the results, these methods exhibit slower accuracy improvements, particularly during the early stages of training. Similarly, FedSpeed [9] also shows slow progress in the initial stages and overall lags behind other methods, failing to demonstrate its full potential.

In summary, FedCM demonstrates superior performance in heterogeneous data environments, as evidenced by the following two key aspects:

- Faster Convergence: FedCM quickly achieves high test accuracy in the early rounds, outperforming other methods in terms of convergence speed.
- Highest Accuracy: FedCM achieves a final accuracy of 0.68, surpassing other methods and demonstrating strong stability and generalization capabilities.

These findings highlight the effectiveness of FedCM in addressing the challenges posed by data heterogeneity, particularly in non-long-tailed scenarios on the CIFAR-10 dataset. The synergy between momentum mechanisms and consensus updates plays a crucial role in improving convergence speed and achieving superior accuracy.

D.2 Supplementary Experiments for FedGrab on Cifar10 Dataset

In this part, We present the experimental results for FedGrab[7] on the Cifar10 dataset, which were not included in the main paper due to space limitations. The experiments are conducted using the official reproduction code provided by the authors. These results further demonstrate the effectiveness of FedGrab on image classification tasks. The supplementary experiments are based on the Cifar10 dataset, a widely-used benchmark for image classification in federated learning. We follow the same experimental settings as described in the main paper, including the number of clients, local epochs, and communication rounds. For completeness, we summarize the key hyperparameters used in Table 3.

As shown in Table 3, we present the performance comparison across different methods (FedAvg, BalanceFL, FedGrab, FedCM and its variants, FedWCM) on the CIFAR-10 dataset under different imbalance factors (IF) and β values of 0.6 and 0.1. Specifically, we focus on the results obtained by FedGrab and compare them with other methods.

From the results, we observe that while FedGrab achieves relatively high accuracy in some cases (e.g., when $IF = 1$, $IF = 0.1$ and $IF = 0.5$), its overall performance is still inferior to FedWCM. FedWCM consistently provides better test accuracy across all imbalance factors, especially in highly imbalanced scenarios ($IF = 0.05$ and $IF = 0.01$), where it significantly outperforms FedGrab.

While FedGrab performs relatively well under $\beta = 0.6$ in some cases, its performance under $\beta = 0.1$ suffers significantly. Specifically, in highly heterogeneous data scenarios (e.g., $IF = 0.1$ and $IF = 0.05$), FedGrab's performance degrades sharply, as shown by the results for $IF = 0.1$, where FedGrab achieves only 32.60% accuracy, compared to 67.75% for FedAvg and 72.07% for FedWCM. This indicates that FedGrab is less effective in handling scenarios with increased data heterogeneity.

In contrast, FedWCM demonstrates robustness across all IF values, particularly under $\beta = 0.1$, where its results remain consistently superior. This highlights the advantage of FedWCM's weighted aggregation mechanism in mitigating the adverse effects of data heterogeneity.

Table 3: Performance comparison on CIFAR-10 under $\beta = 0.6$ and $\beta = 0.1$ with varying imbalance factors (IF). The reported results represent the mean test accuracy across 3 trials using different random seeds.

		FedAvg		BalanceFL		FedGrab		FedCM		+ Focal Loss		+ Balance Loss		+ 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	0.6	0.1
CIFAR-10	1	0.7906	0.6881	0.7629	0.6813	0.7950	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	0.7810	0.6560	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	0.6880	0.3260	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	0.5000	0.1870	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	0.3140	0.1350	0.1865	0.2312	0.1687	0.2023	0.2087	0.2405	0.1249	0.1584	0.4983	0.5012

E Proof of Convergence for FedWCM

E.1 Notations and Definitions

Let $F_0 = \emptyset$ and define $F^i_{r,k} := \sigma(\{x^i_{r,j}\}_{0\le j\le k} \cup F_r)$ and $F_{r+1} := \sigma\left(\bigcup_i F^i_{r,K}\right)$ for all $r \ge 0$, where $\sigma(\cdot)$ denotes the σ-algebra. Let $\mathbb{E}_r[\cdot] := \mathbb{E}[\cdot|F_r]$ represent the expectation conditioned on the filtration F_r, with respect to the random variables $S_r, \{\xi^i_{r,k}\}_{1\le i\le N, 0\le k<K}$ in the r-th iteration. We also use $\mathbb{E}[\cdot]$ to denote the global expectation over all randomness in the algorithms.

For all $r \ge 0$, we define the following auxiliary variable to facilitate the proofs:

$$\epsilon_r := \mathbb{E}\left[\|\nabla f(x_r) - g_{r+1}\|^2\right],$$

where g_{r+1} represents the aggregated gradient at iteration $r+1$.

We further define:

$$U_r := \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}\left[\|x^i_{r,k} - x_r\|^2\right],$$

where $x^i_{r,k}$ denotes the k-th local update of client i during the r-th round and x_r is the global model at round r.

We also introduce:

$$\zeta^i_{r,k} := \mathbb{E}\left[x^i_{r,k+1} - x^i_{r,k} \mid F^i_{r,k}\right],$$

which represents the expected update between successive local updates on client i.

To measure the aggregated local update gradients, we define:

$$\Xi_r := \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\|\zeta^i_{r,0}\|^2\right].$$

Finally, throughout the appendix, let:

$$\Delta := f(x_0) - f^*, \quad G_0 := \frac{1}{N}\sum_{i=1}^{N}\|\nabla f_i(x_0)\|^2,$$

where f^* is the optimal function value, and G_0 represents the initial gradient norm. Additionally, we set $x_{-1} := x_0$ for notational convenience.

E.2 Preliminary Lemmas

Lemma 1. Let $\{X_1, \cdots, X_\tau\} \subset \mathbb{R}^d$ be random variables. If their marginal means and variances satisfy $\mathbb{E}[X_i] = \mu_i$ and $\mathbb{E}[\|X_i - \mu_i\|^2] \le \sigma^2$, then the following inequality holds:

$$\mathbb{E}\left[\left\|\sum_{i=1}^{\tau} X_i\right\|^2\right] \le \left\|\sum_{i=1}^{\tau}\mu_i\right\|^2 + \tau^2\sigma^2.$$

Additionally, if the random variables are correlated in a Markov way such that $\mathbb{E}[X_i \mid X_{i-1}, \cdots, X_1] = \mu_i$ and $\mathbb{E}[\|X_i - \mu_i\|^2] \le \sigma^2$, i.e., the variables $\{X_i - \mu_i\}$ form a martingale, then the following tighter bound applies:

$$\mathbb{E}\left[\left\|\sum_{i=1}^{\tau} X_i\right\|^2\right] \le 2\mathbb{E}\left[\left\|\sum_{i=1}^{\tau}\mu_i\right\|^2\right] + 2\tau\sigma^2.$$

These results are adapted the work of [10].

Lemma 2. Suppose $\{X_1, \cdots, X_\tau\} \subset \mathbb{R}^d$ be random variables that are potentially dependent. If their marginal means and variances satisfy $\mathbb{E}[X_i] = \mu_i$ and $\mathbb{E}[\|X_i - \mu_i\|^2] \le \sigma^2$, then it holds that

$$\mathbb{E}\left[\left\|\sum_{i=1}^{\tau} X_i\right\|^2\right] \le \left\|\sum_{i=1}^{\tau}\mu_i\right\|^2 + \tau^2\sigma^2.$$

If they are correlated in the Markov way such that $\mathbb{E}[X_i \mid X_{i-1}, \cdots, X_1] = \mu_i$ and $\mathbb{E}[\|X_i - \mu_i\|^2] \le \sigma^2$, i.e., the variables $\{X_i - \mu_i\}$ form a martingale, then the following tighter bound holds:

$$\mathbb{E}\left[\left\|\sum_{i=1}^{\tau} X_i\right\|^2\right] \le 2\mathbb{E}\left[\left\|\sum_{i=1}^{\tau}\mu_i\right\|^2\right] + 2\tau\sigma^2.$$

These results follow from Lemma 1 in Scaffold [10].

Lemma 3. Let x_r denote the global model at round r, and let $x^i_{r,k}$ be the local models for client i after k local updates. Assume that the weights w^r_i are computed using the Softmax function based on the deviation of the local data distribution from the global distribution, i.e.,

$$w^r_i = \frac{\exp(s^r_i/T)}{\sum_{j=1}^{N}\exp(s^r_j/T)},$$

where s^r_i measures the deviation of client i's local distribution from the global distribution. Then, the weighted average of the local gradients,

$$\frac{1}{K}\sum_{i=1}^{N}\sum_{k=1}^{K} w^r_i \nabla f(x^i_{r,k}),$$

is closer to the global gradient $\nabla f(x_r)$ than the unweighted average,

$$\frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\nabla f(x^i_{r,k}),$$

in terms of the ℓ_2-norm.

Proof. Define the deviations for each client as:

$$\Delta_i = \nabla f(x_r) - \frac{1}{K}\sum_{k=1}^{K} \nabla f(x_{r,k}^i),$$

and let the set of deviations be $\Delta = (\Delta_1, \Delta_2, \ldots, \Delta_N)$. Denote the weights as $w = (w_1, w_2, \ldots, w_N)$, satisfying $\sum_{i=1}^{N} w_i = 1$ and $w_i \geq 0$. The unweighted average corresponds to uniform weights $w_i = \frac{1}{N}$, and the weighted average uses the Softmax weights.

To prove the inequality, we will explicitly use the inverse relationship between w_i and Δ_i, along with the properties of the rearrangement inequality.

First, sort w_i and Δ_i such that:

$$w_1 \leq w_2 \leq \cdots \leq w_N \quad \text{and} \quad \Delta_1 \geq \Delta_2 \geq \cdots \geq \Delta_N.$$

Under this ordering, the pairwise product $w_i\Delta_i$ is minimized compared to any other pairing of w_i and Δ_i due to the rearrangement inequality. Specifically, for any permutation σ of $\{1, 2, \ldots, N\}$, the following holds:

$$\sum_{i=1}^{N} w_i\Delta_i \leq \sum_{i=1}^{N} w_i\Delta_{\sigma(i)}.$$

Equality is achieved only when Δ_i and w_i are paired in reverse order (i.e., the largest Δ_i is matched with the smallest w_i, and so on).

Next, consider the unweighted arithmetic mean $\frac{1}{N}\sum_{i=1}^{N} \Delta_i$, which corresponds to the case where all weights are uniform ($w_i = \frac{1}{N}$ for all i). For uniform weights, we have:

$$\frac{1}{N}\sum_{i=1}^{N} \Delta_i = \sum_{i=1}^{N} \frac{1}{N}\Delta_i.$$

Now, compare $f(\Delta, w) = \sum_{i=1}^{N} w_i\Delta_i$ to this uniform weighting. By the rearrangement inequality, the weighted sum $f(\Delta, w)$ is minimized when w_i and Δ_i are inversely related (as given in the problem statement). However, for uniform weights, the weights $w_i = \frac{1}{N}$ correspond to the mean value of Δ_i, which is always greater than or equal to the weighted sum $f(\Delta, w)$ when w_i and Δ_i satisfy the inverse relationship:

$$\sum_{i=1}^{N} w_i\Delta_i \leq \sum_{i=1}^{N} \frac{1}{N}\Delta_i.$$

Thus, we have:

$$f(\Delta, w) = \sum_{i=1}^{N} w_i\Delta_i \leq \frac{1}{N}\sum_{i=1}^{N} \Delta_i.$$

Equality holds if and only if all Δ_i are equal, in which case the weighting has no effect. This completes the proof. □ □

E.3 Assumption

Assumption 1. 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

$$\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x - y\|.$$

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

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

and

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

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

E.4 Proof of FedWCM

Lemma 4. If $\gamma L \le \frac{\alpha_r}{6}$, the following holds for $r \ge 1$:

$$\epsilon_r \le \left(1 - \frac{8\alpha_r}{9}\right)\epsilon_{r-1} + \frac{4\gamma^2 L^2}{\alpha_r}\mathbb{E}[\|\nabla f(x_{r-1})\|^2] + \frac{2\alpha_r^2\sigma^2}{NK} + 4\alpha_r L^2 U_r.$$

Additionally, it holds for $r = 0$ that

$$\epsilon_0 \le (1 - \alpha_0)\epsilon_{-1} + \frac{2\alpha_0^2\sigma^2}{NK} + 4\alpha_0 L^2 U_0.$$

Proof. For $r \ge 1$, we have

$$\begin{aligned}\epsilon_r &= \mathbb{E}[\|\nabla f(x_r) - g_{r+1}\|^2] \\ &= \mathbb{E}\left[\left\|(1 - \alpha_r)(\nabla f(x_r) - g_r) + \alpha_r\left(\nabla f(x_r) - \frac{1}{K}\sum_i w_i \sum_k \nabla F(x_{r,k}^i; \xi_{r,k}^i)\right)\right\|^2\right]\end{aligned}$$

Expanding the square, we get

$$\epsilon_r = \mathbb{E}[\|(1 - \alpha_r)(\nabla f(x_r) - g_r)\|^2] + \alpha_r^2\mathbb{E}\left[\left\|\nabla f(x_r) - \frac{1}{K}\sum_{i,k} w_i \nabla F(x_{r,k}^i; \xi_{r,k}^i)\right\|^2\right]$$

$$+2\alpha_r\mathbb{E}\left[\left\langle(1 - \alpha_r)(\nabla f(x_r) - g_r), \nabla f(x_r) - \frac{1}{K}\sum_{i,k} w_i \nabla f(x_{r,k}^i)\right\rangle\right].$$

Note that $\{\nabla F(x_{r,k}^i; \xi_{r,k}^i)\}_{0 \le k < K}$ are sequentially correlated. Using the AM-GM inequality, Lemma 1 and Lemma 3, we have

$$\epsilon_r \le \left(1 + \frac{\alpha_r}{2}\right)\mathbb{E}[\|(1 - \alpha_r)(\nabla f(x_r) - g_r)\|^2] + 2\alpha_r L^2 U_r + 2\alpha_r^2\left(\frac{\sigma^2}{NK} + L^2 U_r\right).$$

Using the AM-GM inequality again and Assumption 1, we obtain

$$\epsilon_r \le (1 - \alpha_r)^2\left(1 + \frac{\alpha_r}{2}\right)\epsilon_{r-1} + \left(1 + \frac{\alpha_r}{2}\right)L^2\mathbb{E}[\|x_r - x_{r-1}\|^2] + 2\alpha_r^2\frac{\sigma^2}{NK} + 4\alpha_r L^2 U_r.$$

Substituting $\|x_r - x_{r-1}\|^2 \le 2\gamma^2(\|\nabla f(x_{r-1})\|^2 + \|g_r - \nabla f(x_{r-1})\|^2)$ and using $\gamma L \le \frac{\alpha_r}{6}$, we get

$$\epsilon_r \le \left(1 - \frac{8\alpha_r}{9}\right)\epsilon_{r-1} + \frac{4\gamma^2 L^2}{\alpha_r}\mathbb{E}[\|\nabla f(x_{r-1})\|^2] + \frac{2\alpha_r^2\sigma^2}{NK} + 4\alpha_r L^2 U_r.$$

Similarly, for $r = 0$,

$$\epsilon_0 \le \left(1 + \frac{\alpha_0}{2}\right)\mathbb{E}[\|(1 - \alpha_0)(\nabla f(x_0) - g_0)\|^2] + 2\alpha_0 L^2 U_0 + 2\alpha_0^2\left(\frac{\sigma^2}{NK} + L^2 U_0\right).$$

Thus, we have

$$\epsilon_0 \le (1 - \alpha_0)\epsilon_{-1} + \frac{2\alpha_0^2\sigma^2}{NK} + 4\alpha_0 L^2 U_0.$$

□

Lemma 5. If $\eta LK \le \frac{1}{\alpha_r}$, the following holds for $r \ge 0$:

$$U_r \le 2eK^2\Xi_r + K\eta^2\alpha_r^2\sigma^2\left(1 + 2K^3L^2\eta^2\alpha_r^2\right).$$

Proof. Recall that $\zeta_{r,k}^i := \mathbb{E}[x_{r,k+1}^i - x_{r,k}^i | F_{r,k}^i] = -\eta((1-\alpha_r)g_r + \alpha_r\nabla f_i(x_{r,k}^i))$. Then we have

$$\mathbb{E}[\|\zeta_{r,j}^i - \zeta_{r,j-1}^i\|^2] \le \eta^2L^2\alpha_r^2\mathbb{E}[\|x_{r,j}^i - x_{r,j-1}^i\|^2] \le \eta^2L^2\alpha_r^2(\eta^2\alpha_r^2\sigma^2 + \mathbb{E}[\|\zeta_{r,j-1}^i\|^2]).$$

For any $1 \le j \le k-1 \le K-2$, using $\eta L \le \frac{1}{\alpha_r K} \le \frac{1}{\alpha_r(k+1)}$, we have

$$\mathbb{E}[\|\zeta_{r,j}^i\|^2] \le \left(1 + \frac{1}{k}\right)\mathbb{E}[\|\zeta_{r,j-1}^i\|^2] + (1+k)L^2\eta^4\alpha_r^4\sigma^2.$$

Unrolling the recursive bound and using $\left(1 + \frac{2}{k}\right)^k \le e^2$, we get

$$\mathbb{E}[\|\zeta_{r,j}^i\|^2] \le e^2\mathbb{E}[\|\zeta_{r,0}^i\|^2] + 4k^2L^2\eta^4\alpha_r^4\sigma^2.$$

By Lemma 2, it holds that for $k \ge 2$,

$$\mathbb{E}[\|x_{r,k}^i - x_r\|^2] \le 2\mathbb{E}\left[\left(\sum_{j=0}^{k-1}\zeta_{r,j}^i\right)^2\right] + 2k\eta^2\alpha_r^2\sigma^2 \le 2e^2k^2\mathbb{E}[\|\zeta_{r,0}^i\|^2] + 2k\eta^2\alpha_r^2\sigma^2\left(1 + 4k^3L^2\eta^2\alpha_r^2\right).$$

This is also valid for $k = 0, 1$. Summing up over i and k finishes the proof. □

Lemma 6. If $288e(\eta KL)^2\left((1-\alpha_r)^2 + e(\alpha_r\gamma LR)^2\right) \le 1$, then it holds for $r \ge 0$ that

$$\sum_{r=0}^{R-1}\Xi_r \le \frac{1}{72eK^2L^2}\sum_{r=-1}^{R-2}(\epsilon_r + \mathbb{E}[\|\nabla f(x_r)\|^2]) + 2\eta^2\alpha_r^2eRG_0.$$

Proof. Note that $\zeta_{r,0}^i = -\eta((1-\alpha_r)g_r + \alpha_r\nabla f_i(x_r))$, so we have

$$\frac{1}{N}\sum_{i=1}^N\|\zeta_{r,0}^i\|^2 \le 2\eta^2\left((1-\alpha_r)^2\|g_r\|^2 + \alpha_r^2\frac{1}{N}\sum_{i=1}^N\|\nabla f_i(x_r)\|^2\right).$$

Using Young's inequality, we have for any $q > 0$ that

$$\mathbb{E}[\|\nabla f_i(x_r)\|^2] \le (1+q)\mathbb{E}[\|\nabla f_i(x_{r-1})\|^2] + (1+q^{-1})L^2\mathbb{E}[\|x_r - x_{r-1}\|^2].$$

Summing over r and applying the upper bound of η completes the proof. □

Theorem E.1. Under Assumptions 1 and 2, if we take $g_0 = 0$, $\alpha_r = \min\left(\sqrt{\frac{NKL\Delta}{\sigma^2R}}, 1\right)$ for any constant $c \in (0, 1]$, $\gamma = \min\left(\frac{1}{24L}, \frac{\alpha_r}{6L}\right)$, and

$$\eta KL \lesssim \min\left(1, \frac{1}{\alpha_r\gamma LR}, \left(\frac{L\Delta}{G_0\alpha_r^3R}\right)^{1/2}, \frac{1}{\sqrt{\alpha_rN}}, \frac{1}{(\alpha_r^3NK)^{1/4}}\right),$$

then FedWCM converges as

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

Here $G_0 := \frac{1}{N}\sum_{i=1}^N\|\nabla f_i(x_0)\|^2$.

Proof. Combining Lemmas 4 and 5, we have

$$\epsilon_r \le \left(1 - \frac{8\alpha_r}{9}\right)\epsilon_{r-1} + 4(\gamma L)^2\frac{1}{\alpha_r}\mathbb{E}[\|\nabla f(x_{r-1})\|^2] + 2\alpha_r^2\frac{\sigma^2}{NK} + 4\alpha_rL^2\left(2eK^2\Xi_r + K\eta^2\alpha_r^2\sigma^2\left(1 + 2K^3L^2\eta^2\alpha_r^2\right)\right),$$

and

$$\epsilon_0 \le (1-\alpha_0)E_{-1} + 2\alpha_0^2\frac{\sigma^2}{NK} + 4\alpha_0L^2\left(2eK^2\Xi_0 + K\eta^2\alpha_0^2\sigma^2\left(1 + 2K^3L^2\eta^2\alpha_0^2\right)\right).$$

Combining these with Lemma 6 and applying the choice of η, γ, and α_0 completes the proof. □

References

[1] Durmus Alp Emre Acar, Yue Zhao, Brendan Rogers, Praneeth Vepakomma, Tian Li Zhu, Matthew Mattina, Vikas Chandra, and Mehdi Joshi. 2021. Federated learning dynamics: Challenges and opportunities. In NeurIPS 2021 Federated Learning Workshop.

[2] 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).

[3] Junfeng Fan and Frederik Vercauteren. 2012. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive (2012).

[4] Ziqing Fan, Shengchao Hu, Jiangchao Yao, Gang Niu, Ya Zhang, Masashi Sugiyama, and Yanfeng Wang. 2024. Locally Estimated Global Perturbations are Better than Local Perturbations for Federated Sharpness-aware Minimization. arXiv preprint arXiv:2405.18890 (2024).

[5] 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.

[6] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In Proceedings of the forty-first annual ACM symposium on Theory of computing. 169–178.

[7] Yukun Guo, Xiaoqiang Ma, Liang Chen, Helong Zhou, Hao Lu, and Xiang-Yang Li. 2022. FedGraB: Addressing Class Imbalance in Federated Learning through Gradient Balancer and Direct Prior Analysis. European Conference on Computer Vision (2022), 733–749.

[8] Abhinav Gupta and Saurabh Kumar. 2021. FedSMOO: Federated Smooth Optimizer for Learning with Heterogeneous Data. arXiv preprint arXiv:2105.08335 (2021).

[9] Farzin Haddadpour, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. 2019. FedSpeed: Efficient Distributed Learning. In Advances in Neural Information Processing Systems.

[10] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In Proceedings of the 37th International Conference on Machine Learning. PMLR.

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

[12] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. In Proceedings of Machine Learning and Systems 2020.

[13] Wanqi Liu, Liangyu Chen, Dengfeng Ke, Yuxin Ding, Yunfeng Gao, Yaqian Li, Haoyu Ma, Heming Zhang, Xiuying Chen, Hui Xue, Tao Qin, Wei Chen, and Tie-Yan Liu. 2023. Classification Re-training Under Label Skew For Long-tailed Federated Learning. European Conference on Computer Vision (2023), 40–56.

[14] 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.

[15] 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.

[16] Yan Sun, Li Shen, Shixiang Chen, Liang Ding, and Dacheng Tao. 2023. Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape. In International Conference on Machine Learning. PMLR, 32991–33013.

[17] 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.

[18] 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).

[19] 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.

[20] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. 2023. Heterogeneous federated learning: State-of-the-art and research challenges. Comput. Surveys 56, 3 (2023), 1–44.

[21] Yu Zhang, Haoyu Ma, Hanze Dong, Xiangyu Zhu, Xiuying Chen, Heming Zhang, Yaqian Li, Hao Zhu, Yunfeng Gao, Xiong Li, and Wanli Ouyang. 2023. CLIP2FL: Enhancing Visual Long-tailed Federated Learning with Diversified Model Initialization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 25009–25018.

[22] Liang Zheng, Siyuan Liu, and Jun Wang. 2020. FedSAM: Federated Sharpness-Aware Minimization. arXiv preprint arXiv:2009.09707 (2020).