# Fed-CBS: A Heterogeneity-Aware Client Sampling Mechanism for Federated Learning via Class-Imbalance Reduction

**Jianyi Zhang** [1] **Ang Li** [2] **Minxue Tang** [1] **Jingwei Sun** [1] **Xiang Chen** [3] **Fan Zhang** [4] **Changyou Chen** [5] **Yiran Chen** [1] **Hai Li** [1]

## Abstract

Due to the often limited communication bandwidth of edge devices, most existing federated learning (FL) methods randomly select only a subset of devices to participate in training at each communication round. Compared with engaging all the available clients, such a random-selection mechanism could lead to significant performance degradation on non-IID (independent and identically distributed) data. In this paper, we present our key observation that the essential reason resulting in such performance degradation is the class-imbalance of the grouped data from randomly selected clients. Based on this observation, we design an efficient heterogeneity-aware client sampling mechanism, namely, Federated Class-balanced Sampling (Fed-CBS), which can effectively reduce class-imbalance of the grouped dataset from the intentionally selected clients. We first propose a measure of class-imbalance which can be derived in a privacy-preserving way. Based on this measure, we design a computation-efficient client sampling strategy such that the actively selected clients will generate a more class-balanced grouped dataset with theoretical guarantees. Experimental results show that Fed-CBS outperforms the status quo approaches in terms of test accuracy and the rate of convergence while achieving comparable or even better performance than the ideal setting where all the available clients participate in the FL training.

[1]Duke University [2]University of Maryland, College Park [3]George Mason University [4]Yale University [5]The State University of New York at Buffalo. Correspondence to: Jianyi Zhang <jianyi.zhang@duke.edu>.

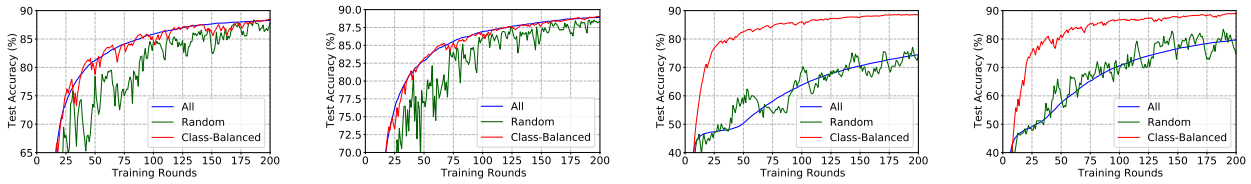
*Proceedings of the $40^{th}$ International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023. 

## 1. Introduction

With the booming of IoT devices, a considerable amount of data is generated at the network edge, providing valuable resources for learning insightful information and enabling intelligent applications such as self-driving, video analytics, anomaly detection, etc. The traditional wisdom is to train machine learning models by collecting data from devices and performing centralized training. Data migration usually raises serious privacy concerns. Federated learning (FL) (McMahan et al., 2017a) is a promising technique to mitigate such privacy concerns, enabling a large number of clients to learn a shared model collaboratively, and the learning process is orchestrated by a central server. In particular, the participating clients first download a global model from the central server and then compute local model updates using their local data. The clients then transmit the local updates to the server, where the local updates are aggregated and then the global model is updated accordingly.

In practice, due to limited communication and computing capabilities, one usually can not engage all the available clients in FL training to fully utilize all the local data. Therefore, most FL methods only randomly select a subset of the available clients to participate in the training in each communication round. However, in practice, the data held by different clients are often typically non-IID (independent and identically distributed) due to various user preferences and usage patterns. This leads to a serious problem that the random client selection strategy often fails to learn a global model that can generalize well for most of the participating clients under non-IID settings (Goetz et al., 2019; Cho et al., 2020; Nishio & Yonetani, 2019; Yang et al., 2020).

Several heuristic client selection mechanisms have been proposed to tackle the non-IID challenge. For example, in the method of (Goetz et al., 2019), the clients with larger local loss will have a higher probability to be selected to participate in the training. Power-of-Choice (Cho et al., 2020) selects several clients with the largest loss from a randomly sampled subset of all the available clients. However, selecting clients with a larger local loss may not guarantee that the final model can have a smaller global loss. Another limitation of previous research on client selection is




(a) Global Balanced & One-class (b) Global Balanced & Two-class (c) Global Imbalanced & One-class (d) Global Imbalanced & Two-class

*Figure 1.* Three different FL client selection strategies on MNIST. *All* means engaging all the 100 clients in training. *Random* means randomly selecting 10 clients. *Class Balanced* means that we keep the class-balance by intentionally selecting 10 clients. In Figure 1a and 1b, the global dataset of all the 100 clients' training data is class-balanced. In Figure 1c and 1d, the global dataset is class-imbalanced. Each client has only one class of data in (a) and (c) and each client has two classes of data in (b) and (d). The results show significant performance degradation with imbalanced data from random client selection. It is worth noting that when the global dataset is class-imbalanced, selecting all the clients leads to worse performance compared with the *Class Balanced* strategy, which suggests the importance of keeping class-balance for client selection.

the missing comparison between their strategy and the ideal case, where all the available clients participate in the training. In general, existing works not only miss a vital criterion that can measure the performance of their methods, but also fail to investigate the essential reason why random client selection can lead to performance degradation on non-IID data compared with fully engaging all the available clients.

In this paper, we focus on image classification tasks. First, we demonstrate our key observation for the essential reason why random client selection results in performance degradation on non-IID data, which is the *class-imbalance* of the grouped dataset from randomly selected clients. Based on our observation, we design an efficient heterogeneity-aware client sampling mechanism, *i.e.*, Federated Class-Balanced Sampling (Fed-CBS), which effectively reduces the class-imbalance in FL. Fed-CBS is orthogonal to numerous existing techniques to improve the performance of FL (Li et al., 2018; Wang et al., 2020b; Karimireddy et al., 2019; Chen et al., 2020; Reddi et al., 2020; Hao et al., 2021; Yang et al., 2021) on non-IID data, meaning Fed-CBS can be integrated with these methods to improve their performance further. Our major contributions are summarized as follows:

- We reveal that the class-imbalance is the fundamental reason why random client selection leads to performance degradation on non-IID data in Section 2.
- To effectively reduce the class-imbalance, we design an efficient heterogeneity-aware client sampling mechanism, *i.e.*, Fed-CBS, based on our proposed class-imbalance metric in Section 3. We provide theoretical analysis on the convergence of Fed-CBS in Section 4, as well as the analysis of the NP-hardness of this problem.
- We empirically evaluate Fed-CBS on FL benchmark (non-IID datasets) in Section 5. The results demonstrate that Fed-CBS can improve the accuracy of FL models on CIFAR-10 by $2\% \sim 7\%$ and accelerate the convergence time by $1.3\times \sim 2.8\times$, compared with the state-of-the-art method (Yang et al., 2020) that also aims to reduce class-imbalance via client selection. Furthermore, our Fed-CBS achieves comparable or even better performance than the ideal setting where all the available devices are involved in the training.

## 2. Preliminary and Related Work

We first clarify three definitions. The **local dataset** is the client's own locally-stored dataset, which is inaccessible to other clients and the server. Due to the heterogeneity of local data distribution, the phenomenon of class-imbalance frequently happens in most of the local datasets. The **global dataset** is the union of all the available client local datasets. It can be class-balanced or class-imbalanced, but it is often imbalanced. The **grouped dataset** is the union of several clients' local datasets which have been selected to participate in training for one communication round. It follows that the grouped dataset is a subset of the global dataset.

### 2.1. Pitfall of Class-Imbalance in Client Selection

Some recent works (Yang et al., 2020; Wang et al., 2020b; Duan et al., 2019) have identified the issue of class-imbalance in the grouped dataset by random selection under non-IID settings. Since class-imbalance degrades the classification accuracy on minority classes (Huang et al., 2016) and leads to low training efficiency, we are motivated to verify whether the class-imbalance of the randomly-selected grouped dataset is the essential reason accounting for the performance degradation.

We conduct some experiments on MNIST to verify our proposition[1]. As shown in Figure 1a and Figure 1b, the random selection mechanism shows the worst performance when the global label distribution is class-balanced. If we keep the grouped dataset class-balanced by manually selecting the clients based on their local label distribution, we can

[1]Detailed experiment settings are listed in the Appendix (Section C.1)

obtain accuracy comparable to the case of fully engaging all the clients in training.

Another natural corollary is that when the global dataset is inherently class-imbalanced, engaging all clients in training may lead to worse performance than manually keeping the grouped dataset class-balanced. The results in Figure 1c and Figure 1d prove our hypothesis and verify the importance of class-imbalance reduction. This also indicates that only keeping diversity in the data and fairness for clients is not enough, which was missed in the previous literature (Balakrishnan et al., 2021; Huang et al., 2021; Yang et al., 2020; Wang et al., 2020b; Shen et al., 2022; Wang et al., 2021). More experimental results on larger datasets will be provided to verify the importance of class-imbalance reduction (Section 5).

### 2.2. Related Work

Some effort has been made to improve client selection for FL in previous literature. (Cho et al., 2020; Goetz et al., 2019) select clients with larger local loss, but this cannot guarantee that the final global model has a smaller global loss. Focusing on the diversity in client selection, the authors of (Balakrishnan et al., 2021) select clients by maximizing a submodular facility location function defined over gradient space. A fairness-guaranteed algorithm termed RBCS-F was proposed in (Huang et al., 2021), which models the fairness-guaranteed client selection as a Lyapunov optimization problem. Although diversity and fairness are important, the experimental results in Section 2.1 demonstrate that they are not enough for client selection if the class-imbalance issue is not considered. The authors in (Ribero & Vikalo, 2020) model the progression of model weights by an Ornstein-Uhlenbeck process and design a sampling strategy for selecting clients with significant weight updates. However, the work only considers the identical data distribution setting. Following the existing works (Goetz et al., 2019; Cho et al., 2020), we only focus on the data heterogeneity caused by non-IID data across clients. Additionally, we included a comparison of our method with other clustered-based client sampling algorithms in the appendix.

To the best of our knowledge, (Duan et al., 2019) and (Yang et al., 2020) are the first two attempts to improve client selection by reducing class-imbalance. An extra virtual component called a mediator is introduced in *Astraea* of (Duan et al., 2019), which has access to the local label distributions of the clients. With these distributions, *Astraea* will conduct client selection in a greedy way. The method of (Yang et al., 2020) first estimates the local label distribution of each client based on the gradient of model parameters and adopts the same greedy way to select clients as *Astraea*. Since directly knowing the exact value of local label distributions of clients in *Astraea* will cause severe concerns on privacy leakage, we consider the method in (Yang et al., 2020) as the state-of-the-art method aiming to improve client selection through class-imbalance reduction.

However, the solution presented by (Yang et al., 2020) has several limitations. First, their method requires a class-balanced auxiliary dataset that consists of all classes of data at the server. However, that is not always available in some large-scale FL systems since it requires the server to collect raw data from clients, which breaches privacy. Second, their estimations of the clients' local label distribution are not accurate as shown in Figure 2. Theorem 1 in (Yang et al., 2020) supports their estimations, but it cannot be generalized to multi-class classification tasks since it has only been proved in the original paper (Anand et al., 1993) for two-class classification problems. Finally, the performance of greedily conducting the client selection is not guaranteed due to the nature of the greedy algorithm. We provide an example in Figure 3 to show its weakness. Their method will select $C_1$ as the first client since it is the most class-balanced one. Then $C_2$ will be selected because the grouped dataset of $C_1 \cup C_2$ is the most class-balanced among the choices $C_1 \cup C_2$, $C_1 \cup C_3$ and $C_1 \cup C_4$. Similarly, it will choose $C_3$ since the grouped dataset of $C_1 \cup C_2 \cup C_3$ is more class-balanced than $C_1 \cup C_2 \cup C_4$. Their method is deterministic and thus only one combination $\{C_1, C_2, C_3\}$ is obtained. However, this is clearly not the optimal solution since $\{C_1, C_3, C_4\}$ is more class-balanced than $\{C_1, C_2, C_3\}$. The above weaknesses motivate us to design a more effective solution for this problem.

## 3. Methodology

We first propose a metric to measure class-imbalance in Section 3.1. Then we derive the measure with privacy-preserving techniques in Section 3.2. Based on this measure, we then design our client sampling mechanism and show its superiority in Section 3.3.

### 3.1. Class-Imbalance Measure

Assume there are $B$ classes of data in an image classification task, where $B \geq 2$. In the $k$-th communication round, we assume there are $N_k$ available clients and we select $M$ clients from them. To make the presentation concise, we ignore the index "k" and assume the set of indices for the available clients is $\{1, 2, 3, ..., N\}$ and the $n$-th available client has its own training dataset $\mathcal{D}_n$. We adopt the following vector of size $B$ to represent the local label distribution of $\mathcal{D}_n$, where $\alpha_{(n,b)} \geq 0$ and $\sum_{b=1}^{B} \alpha_{(n,b)} = 1$,

$$\boldsymbol{\alpha}_n = \left[\alpha_{(n,1)}, \alpha_{(n,2)}, ..., \alpha_{(n,b)}, ..., \alpha_{(n,B)}\right] . \quad (1)$$

We aim to find a subset $\mathcal{M}$ of $\{1, 2, 3, .., N\}$ of size $M$, such that the following grouped dataset $\mathcal{D}^g_{\mathcal{M}} = \bigcup_{n \in \mathcal{M}} \mathcal{D}_n$ is class-balanced. Assuming the $n$-th client's local dataset has

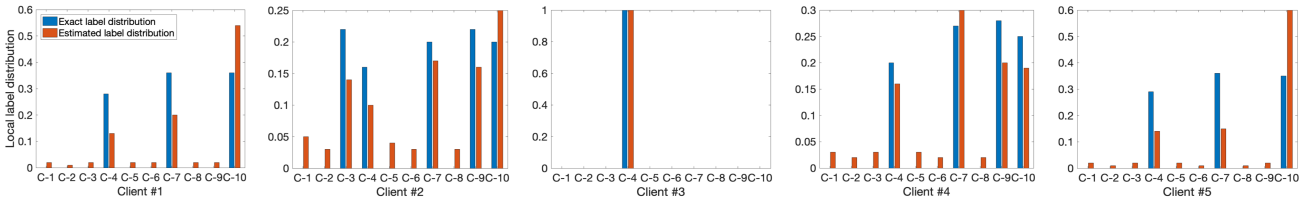
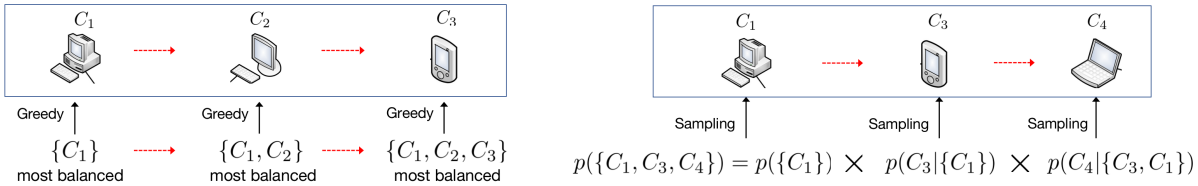



*Figure 2.* The exact local label distributions and the estimated ones of the first 5 clients in the experiment of (Yang et al., 2020). Label distribution quantifies the ratio between the number of data from 10 classes (C-1, C-2, ..., C-10) in each client's local dataset.




(a) Greedy method

(b) Our method

*Figure 3.* An example demonstrating the weakness of greedy method to deal with class imbalance. Supposing we work on a 6-class classification task and aim to select 3 clients from 4 available clients $C_1, C_2, C_3, C_4$. Each of them has 30 images. The compositions of their local datasets are $[5, 5, 5, 5, 5, 5]$, $[6, 6, 6, 6, 6, 0]$, $[0, 0, 0, 10, 10, 10]$ and $[10, 10, 10, 0, 0, 0]$ respectively. The greedy method in (Yang et al., 2020) is deterministic. It can only derive one result $\{C_1, C_2, C_3\}$ instead of the optimal solution $\{C_1, C_3, C_4\}$ (see the text description). But our method is based on probability modeling, which directly models the distribution of the optimal solution $\{C_1, C_3, C_4\}$. Thus when sampling from it, the optimal solution can be returned with high probability.

$q_n$ training samples, the following vector $\boldsymbol{\alpha}^g_{\mathcal{M}}$ can represent the label distribution of the grouped dataset $\mathcal{D}^g_{\mathcal{M}}$,

$$\boldsymbol{\alpha}^g_{\mathcal{M}} = \frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{\alpha}_n}{\sum_{n\in\mathcal{M}} q_n} = \left[\frac{\sum_{n\in\mathcal{M}} q_n \alpha_{(n,1)}}{\sum_{n\in\mathcal{M}} q_n}, ..., \frac{\sum_{n\in\mathcal{M}} q_n \alpha_{(n,b)}}{\sum_{n\in\mathcal{M}} q_n}, ..., \frac{\sum_{n\in\mathcal{M}} q_n \alpha_{(n,B)}}{\sum_{n\in\mathcal{M}} q_n}\right].$$

Instead of dealing with the Kullback-Leibler (KL) divergence as (Duan et al., 2019; Yang et al., 2020), which is complicated to analyze, we propose the following function to measure the magnitude of class-imbalance of $\mathcal{M}$, which we call *Quadratic Class-Imbalance Degree (QCID)*:

$$QCID(\mathcal{M}) \triangleq \sum_{b=1}^{B} \left(\frac{\sum_{n\in\mathcal{M}} q_n \alpha_{(n,b)}}{\sum_{n\in\mathcal{M}} q_n} - \frac{1}{B}\right)^2.$$

Essentially, $QCID(\mathcal{M})$ reflects the $L_2$ distance between the distribution of the grouped dataset $\mathcal{D}^g_{\mathcal{M}}$ and the ideally class-balanced dataset that has a uniform label distribution. Although there exist several more commonly-used probabilistic distances other than $L_2$, it is easier to analyze *QCID* and more efficient to calculate while keeping privacy as shown in the next section.

### 3.2. Privacy-Preserving QCID Derivation

Our privacy goal is to calculate the value of *QCID* while keeping clients' local distributions $\{\boldsymbol{\alpha}_n\}$ hidden from the server since it contains sensitive information. Unlike Kullback-Leibler (KL) divergence which is difficult to analyze, we can expand the expression of *QCID* to explore how the pairwise relationships of the clients' local label distributions $\{\boldsymbol{\alpha}_m\}$ affects the class-imbalance degree of $\mathcal{M}$, where $m \in \mathcal{M}$. Below we provide a theorem to show the feasibility of our method.

**Theorem 3.1.** *The QCID value is decided by the sum of inner products between each two vectors $\boldsymbol{\alpha}_m, \boldsymbol{\alpha}_{m'} \in \{\boldsymbol{\alpha}_m\}$ with $m \in \mathcal{M}$, i.e.,*

$$QCID(\mathcal{M}) = \frac{\sum_{n\in\mathcal{M}, n'\in\mathcal{M}} q_n q'_n \boldsymbol{\alpha}_n \boldsymbol{\alpha}^T_{n'}}{(\sum_{n\in\mathcal{M}} q_n)^2} - \frac{1}{B}$$

Theorem 3.1 reveals the fact that there is no need to know the local label distribution of each client to calculate the *QCID*, as long as we have access to the inner products between each other. To derive the *QCID* for any subset $\mathcal{M} \subseteq \{1, 2, 3, .., N\}$, we only need to know the following $N \times N$ matrix $\boldsymbol{S}$ with element $s_{n,n'}$ being $\boldsymbol{\alpha}_n \boldsymbol{\alpha}^T_{n'}$, which is the inner product between the local label distributions of the available clients $n$ and $n'$.

$$\boldsymbol{S} = \begin{bmatrix} q_1 q_1 \boldsymbol{\alpha}_1 \boldsymbol{\alpha}^T_1 & q_1 q_2 \boldsymbol{\alpha}_1 \boldsymbol{\alpha}^T_2 & \cdots & q_1 q_N \boldsymbol{\alpha}_1 \boldsymbol{\alpha}^T_N \\ q_2 q_1 \boldsymbol{\alpha}_2 \boldsymbol{\alpha}^T_1 & q_2 q_2 \boldsymbol{\alpha}_2 \boldsymbol{\alpha}^T_2 & \cdots & q_2 q_N \boldsymbol{\alpha}_2 \boldsymbol{\alpha}^T_N \\ \vdots & \vdots & \ddots & \vdots \\ q_N q_1 \boldsymbol{\alpha}_N \boldsymbol{\alpha}^T_1 & q_N q_2 \boldsymbol{\alpha}_N \boldsymbol{\alpha}^T_2 & \cdots & q_N q_N \boldsymbol{\alpha}_N \boldsymbol{\alpha}^T_N \end{bmatrix}$$

Although it is possible to calculate *QCID* with $\boldsymbol{S}$, another concern arises, *can a malicious party infer the values of $\{\boldsymbol{\alpha_i}\}$ from $\boldsymbol{S}$ ?* Then we have another theorem to provide privacy protection.

**Theorem 3.2.** *One can not derive the values of $\{\boldsymbol{\alpha_i}\}$ from the value of $\boldsymbol{S}$.*

Based on these two theorems, our privacy goal can be simplified as enabling the server to derive $\boldsymbol{S}$ without access to $\{\boldsymbol{\alpha_i}\}$. There are several ways to achieve our goal. One option is to leverage the server-side trusted execution environments (TEEs), e.g., Intel SGX (Anati et al., 2013), which allows calculating $S$ without leaking information of $\{\alpha_n\}$. Another potential solution is to adopt Fully Homomorphic Encryption (FHE) (Chen et al., 2017; Brakerski et al., 2014; Fan & Vercauteren, 2012; Halevi & Shoup, 2014; 2015) to enable the server to compute on encrypted data (*i.e.*, $\{\boldsymbol{\alpha_i}\}$) to derive $\boldsymbol{S}$. We provide an example of the system skelon in Section A.2 to illustrate how to derive $S$ without knowing the local label distributions $\{\boldsymbol{\alpha_i}\}$ using FHE. Since we focus on efficient algorithms to reduce class-imbalance instead of designing the fundamental infrastructure for computing (which is beyond our scope and not a contribution of this paper), we leave the detailed system design for future work.

### 3.3. A Client Sampling Mechanism

To select the most class-balanced grouped dataset $\mathcal{D}^g_{\mathcal{M}}$, we need to find the optimal subset $\mathcal{M}^*$ that has the lowest $QCID$ value, which is defined as follows:

$$\mathcal{M}^* \triangleq \underset{\mathcal{M}\subseteq\{1,2,3,..,N\}}{\arg\min} \frac{\sum_{n\in\mathcal{M},n'\in\mathcal{M}} q_n q_{n'} \boldsymbol{\alpha}_n \boldsymbol{\alpha}_{n'}^T}{(\sum_{n\in\mathcal{M}} q_n)^2} - \frac{1}{B}.$$

The main challenge is computational complexity. To find the exact optimal $\mathcal{M}^*$, we need to loop through all the possible cases and find the lowest $QCID$ value. The computational complexity thereafter will be $\mathcal{O}\left(\binom{N}{M} \times M^2\right)$, which is unacceptable when $N$ is extremely large.

**A probability approach** To overcome the computational bottleneck, instead of treating $\mathcal{M}$ as a determined set, we consider it as a sequence of random variables, *i.e.* $\mathcal{M} = \{C_1, C_2, ..., C_m, ..., C_M\}$ and assign it with some probability. Our expectation is that $\mathcal{M}$ should have higher probability to be sampled with if it is more class-balanced. This means $P(C_1 = c_1, C_2 = c_2, ..., C_m = c_m, ..., C_M = c_M)$ should be larger if $\mathcal{M} = \{c_1, c_2, ..., c_M\}$ has a lower $QCID$ value. Our sampling strategy generates the elements in $\mathcal{M}$ in a sequential manner, *i.e.*, we first sample $\mathcal{M}_1 = \{c_1\}$ according to the probability of $P(C_1 = c_1)$, then sample $c_2$ to form $\mathcal{M}_2 = \{c_1, c_2\}$ according to the conditional probability $P(C_2 = c_2|C_1 = c_1)$. The same procedure applies for the following clients until we finally obtain $\mathcal{M} = \{c_1, c_2, ..., c_M\}$. In the following, we will design proper conditional probabilities such that the joint distribution of client selection satisfies our expectations.

Let $T_n$ denote the number of times that client $n$ has been selected. Once client $n$ has been selected in a communication round, $T_n \to T_n + 1$, otherwise, $T_n \to T_n$. Inspired by combinatorial upper confidence bounds (CUCB) algorithm (Chen et al., 2013) and previous work in (Yang et al., 2020), in the $k$-th communication round, the first element is designed to be sampled with the following probability:

$$P(C_1 = c_1) \propto \frac{1}{[QCID(\mathcal{M}_1)]^{\beta_1}} + \lambda\sqrt{\frac{3\ln k}{2T_{c_1}}}, \quad \beta_1 > 0,$$

where $\lambda$ above is the exploration factor to balance the trade-off between exploitation and exploration. The second term will add a higher probability to the clients that have never been sampled before in the following communication rounds. After sampling $C_1$, the second client is defined to be sampled with probability

$$P(C_2 = c_2|C_1 = c_1) \propto \frac{\frac{1}{[QCID(\mathcal{M}_2)]^{\beta_2}}}{\frac{1}{[QCID(\mathcal{M}_1)]^{\beta_1}} + \alpha\sqrt{\frac{3\ln k}{2T_{c_1}}}}, \beta_2 > 0.$$

For the m-th client, where $2 < m \leq M$, we define

$$P(C_m = c_m|C_{m-1} = c_{m-1}, ..., C_2 = c_2, C_1 = c_1) \propto \frac{[QCID(\mathcal{M}_{m-1})]^{\beta_{m-1}}}{[QCID(\mathcal{M}_m)]^{\beta_m}}, \quad \beta_{m-1}, \beta_m > 0.$$

With the above sampling process, the final probability to sample $\mathcal{M}$ is $P(C_1 = c_1, C_2 = c_1, ..., C_M = c_M) = P(C_1 = c_1) \times P(C_2 = c_2|C_1 = c_1) \cdots \times P(C_M = c_M|C_{M-1} = c_{M-1}, ..., C_2 = c_2, C_1 = c_1) \propto 1/[QCID(\mathcal{M})]^{\beta_M}$. Since $\beta_M > 0$, this matches our goal that the $\mathcal{M}$ with lower $QCID$ value should have higher probability to be sampled with. Our mechanism, Fed-CBS, is summarized in Algorithm 1 .

**Algorithm 1** Fed-CBS

**Initialization:** initial local model $\boldsymbol{w}^{(0)}$, client index subset $\mathcal{M} = \varnothing$, $K$ communication rounds, $k = 0, T_n = 1$
**while** $k < K$ **do**
  **// Client Selction**:
  **for** $n$ **in** $\{1, 2, ..., N\}$ **do**
    **if** $n \in \mathcal{M}$ **then**
      $T_n \to T_n + 1$
    **else**
      $T_n \to T_n$;
    **end if**
  **end for**
  Update $\mathcal{M}$ using our proposed sampling strategy in Section 3.3
  **// Local Updates:**
  **for** $n \in \mathcal{M}$ **do**
    $\boldsymbol{w}_n^{(k+1)} \leftarrow Update(\boldsymbol{w}^{(k)})$.
  **end for**
  **// Global Aggregation**:
  $\boldsymbol{w}^{(k+1)} \leftarrow Aggregate(\boldsymbol{w}_n^{(k+1)})$ for $n \in \mathcal{M}$
**end while**

**Details and analysis** For any $1 < m < M$, we have

$$P(C_1 = c_1, C_2 = c_1, ..., C_m = c_m) \propto \frac{1}{[QCID(\mathcal{M}_m)]^{\beta_m}}.$$

This means when we generate the first $m$ elements of $\mathcal{M}$, we expect the $\mathcal{M}_m$ should be more class-balanced since the $\mathcal{M}_m$ with lower $QCID$ value has a higher probability of being sampled. This is different from the algorithm in (Yang et al., 2020), which greedily chooses the $c_m$ from $\{1, 2, .., N\}/\mathcal{M}_{m-1}$ that makes $\mathcal{M}_m$ the most class-balanced one. Unlike the greedy algorithm which has no guarantees on finding the optimal client set, our method can generate the globally optimal set of clients in the sense of probability. An example is provided in Figure 3 to demonstrate that our method can overcome the pitfall of the greedy method. After selecting the first two clients, $\{C_1, C_3\}$ our method is less class-balanced than $\{C_1, C_2\}$ chosen by the greedy method. However, after making the last choice, our method has the chance to derive a perfectly class-balanced set $\{C_1, C_3, C_4\}$. In contrast, the greedy method can only get one result $\{C_1, C_2, C_3\}$, which is less class-balanced.

We require the distribution of $P(C_1 = c_1, C_2 = c_1, ..., C_m = c_m)$ to be more dispersed when $m$ is small. This is because we expect our sampling strategy to explore more possible cases of client composition at the beginning. We require the distribution of $P(C_1 = c_1, C_2 = c_1, ..., C_m = c_m)$ to be less dispersed when $m$ is large. This is because as we approach the end of our sampling process, we expect our sampling strategy can find the $\mathcal{M}_m$ that is more class-balanced. Especially when $m = M$, we hope the strategy to find the client $c_M$ which can make $\mathcal{M}$ the most class-balanced. Since

$$P(C_1 = c_1, C_2 = c_1, ..., C_m = c_m) \propto \frac{1}{[QCID(\mathcal{M}_m)]^{\beta_m}}$$

we can set $0 < \beta_1 < \beta_2 < ... < \beta_M$ to satisfy the above requirements.

*Remark: We set a lower bound for $QCID(\mathcal{M}_m)$ as $L_b$ since $QCID(\mathcal{M}_m) = 0$ in some special cases will cause $P(C_m = c_m | C_{m-1} = c_{m-1}, ..., C_1 = c_1) \to \infty$.* When viewing the conditional distribution as the likelihood in Bayesian inference, our probability can be interpreted as an estimate of the posterior distribution. This allows us to comprehend our algorithm through the lens of Bayesian sampling (Welling & Teh, 2011; Liu & Wang, 2019; Zhang et al., 2020a; 2019). In our future studies, we will further analyze the connection between them. Below we present two theorems to show the superiority of our proposed sampling strategy.

**Theorem 3.3** (Class-Imbalance Reduction)**.** *We denote the probability of selecting $\mathcal{M}$ with our strategy with $\beta_M$ as $P_{\beta_M}$ and the probability of selecting $\mathcal{M}$ with the random selection as $P_{rand}$. Our method can reduce the expectation of $QCID$ compared to the random selection mechanism. In other words, we have*

$$\mathbb{E}_{\mathcal{M} \sim P_{\beta_M}} QCID(\mathcal{M}) < \mathbb{E}_{\mathcal{M} \sim P_{rand}} QCID(\mathcal{M}).$$

*Furthermore, if increasing the value $\beta_M$, the expectation of $QCID$ can be further reduced, i.e., for $\beta'_M > \beta_M$, we have*

$$\mathbb{E}_{\mathcal{M} \sim P_{\beta'_M}} QCID(\mathcal{M}) < \mathbb{E}_{\mathcal{M} \sim P_{\beta_M}} QCID(\mathcal{M}).$$

**Theorem 3.4** (Computation Complexity Reduction)**.** *The computation complexity of our method is $\mathcal{O}\left(N \times M^2\right)$, which is much smaller than the exhaustive search of $\mathcal{O}\left(\binom{N}{M} \times M^2\right)$.*

Theorem 3.4 shows that the computation complexity of our method is independent of the number of classes. Since the dimension of neural networks is typically much larger than the class distribution vector $\boldsymbol{\alpha}_n$, the additional communication cost is almost negligible. Besides, we also prove the NP-hardness of the problem formally in Section B.3 in the appendix.

## 4. Convergence Analysis

To analyze the convergence of our method, we first define our objective functions and adopt some general assumptions. Our global objective function $\widetilde{F} > 0$ can be decomposed as $\widetilde{F} = \frac{1}{B}\sum_{b=1}^{B} \widetilde{F}_b$, where $\widetilde{F}_b$ is the averaged loss function with respect to all the data of the $b$-th class in the global dataset. Similarly, the $n$-th client's local objective function $F_n$ can be decomposed as $F_n = \sum_{b=1}^{B} \alpha_{(n,b)} F_{n,b}$, where $F_{n,b}$ is the averaged loss function with respect to all the data of the $b$-th class in the $n$-th client's local dataset, and $\alpha_{(n,b)}$ is defined in Equation 1. Moreover, let $\boldsymbol{w}^{(k)}$ denote the global model parameters at the $k$-th communication round and $\boldsymbol{w}^{(0)}$ denote the initial global model parameters. If not stated explicitly, $\nabla$ denotes $\nabla_w$ throughout the paper.

**Assumption 4.1** (Smoothness)**.** The global objective function $\widetilde{F}$ and each client's averaged loss function $F_{n,b}$ are Lipschitz smooth, *i.e.* $\left\|\nabla \widetilde{F}(\boldsymbol{w}) - \nabla \widetilde{F}(\boldsymbol{w}')\right\| \leq L_{\widetilde{F}} \|\boldsymbol{w} - \boldsymbol{w}'\|$ and $\|\nabla F_{n,b}(\boldsymbol{w}) - \nabla F_{n,b}(\boldsymbol{w}')\| \leq L_{n,b} \|\boldsymbol{w} - \boldsymbol{w}'\|, \forall n, b, \boldsymbol{w}, \boldsymbol{w}'$.

**Assumption 4.2** (Unbiased Gradient and Bounded Variance)**.** The stochastic gradient $g_n$ at each client is an unbiased estimator of the local gradient: $\mathbb{E}_\xi\left[g_n(\boldsymbol{w} \mid \xi)\right] = \nabla F_n(\boldsymbol{w})$, with bounded variance $\mathbb{E}_\xi\left[\|g_n(\boldsymbol{w} \mid \xi) - \nabla F_n(\boldsymbol{w})\|^2\right] \leq \sigma^2, \forall \boldsymbol{w}$, where $\sigma^2 \geq 0$.

**Assumption 4.3** (Bounded Dissimilarity)**.** There exist two non-negative constants $\delta \geq 1, \gamma^2 \geq 0$ such that $\sum_{b=1}^{B} \frac{1}{B} \left\|\nabla \widetilde{F}_b(\boldsymbol{w})\right\|^2 \leq \delta \left\|\sum_{b=1}^{B} \frac{1}{B} \nabla \widetilde{F}_b(\boldsymbol{w})\right\|^2 + \gamma^2, \forall \boldsymbol{w}$.

**Assumption 4.4** (Class-wise Similarity)**.** For each class $b$, the discrepancy between the gradient of global averaged

| | | all | rand | pow-d | Fed-cucb | Fed-CBS |
|---|---|---|---|---|---|---|
| Communication Rounds | $\alpha$=0.1 | 757±155 | 951±202 | 1147±130 | 861±328 | **654±96** |
| | $\alpha$=0.2 | 746±95 | 762±105 | 741±111 | 803±220 | **475±110** |
| | $\alpha$=0.5 | 426±67 | 537±115 | 579±140 | 1080±309 | **384±74** |
| $\mathbb{E}[QCID](10^{-2})$ | $\alpha$=0.1 | 1.01±0.01 | 8.20±0.21 | 12.36±0.26 | 7.09±2.27 | **0.62±0.20** |
| | $\alpha$=0.2 | 0.93±0.03 | 7.54±0.27 | 10.6±0.48 | 5.93±1.01 | **0.51±0.12** |
| | $\alpha$=0.5 | 0.72±0.03 | 5.87±0.24 | 7.36±0.57 | 6.47±0.77 | **0.36±0.04** |

*Table 1.* The communication rounds required for targeted test accuracy and the averaged QCID values. The targeted test accuracy is $45\%$ for $\alpha = 0.1$, $47\%$ for $\alpha = 0.2$ and $50\%$ for $\alpha = 0.5$. The results are the mean and the standard deviation over 4 different random seeds.

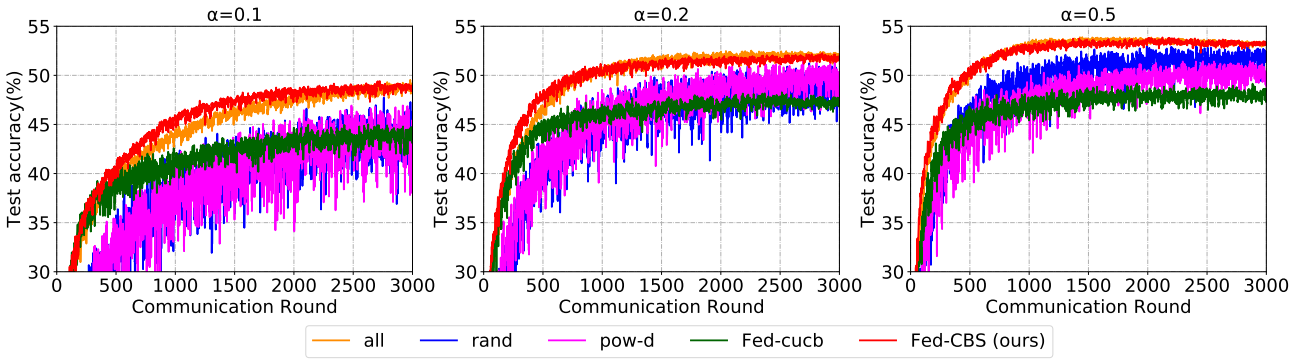



*Figure 4.* Test accuracy on Cifar-10 under three heterogeneous settings.

loss function and the local one is bounded by some constant in $l^2$ norm. That means, for every $n$ and $b$, we have $\left\|\nabla \widetilde{F}_b(\boldsymbol{w}) - \nabla F_{n,b}(\boldsymbol{w})\right\|^2 \leq \kappa_{n,b}^2, \forall \boldsymbol{w}$.

Assumptions 4.1, 4.2 and 4.3 have been widely adopted in previous literature on the theoretical analysis of FL (Li et al., 2019; Cho et al., 2020; Wang et al., 2020a). Assumption 4.4 is based on the similarity among the data from the same class. Similar to the standard setting (Wang et al., 2020a), the convergence of our algorithm is measured by the norm of the gradients, stated in Theorem 4.5.

**Theorem 4.5.** *Under Assumptions 4.1 to 4.4, if the total communication rounds $K$ is pre-determined and the learning rate is set as $\eta = \frac{s}{10L\sqrt{\tau(\tau-1)K}}$, where $s < 1$, $L = \max_{\{n,b\}} L_{n,b}$ and $\tau$ is the number of local update iterations, the minimal gradient norm of $\widetilde{F}$ is bounded as:*

$$\min_{k\leq K} \left\|\nabla \widetilde{F}\left(\boldsymbol{w}^{(k)}\right)\right\|^2 \leq \frac{1}{V}\Big[\frac{\sigma^2 s^2}{25\tau K} + \frac{sL_{\widetilde{F}}\sigma^2}{10L\sqrt{\tau(\tau-1)K}} + 5\kappa^2 + \frac{10L\sqrt{\tau(\tau-1)}\widetilde{F}\left(\boldsymbol{w}^{(0)}\right)}{s\sqrt{K}} + \gamma^2\mathbb{E}[QCID]\Big],$$

*where $V = \frac{1}{3} - \delta B\mathbb{E}[QCID]$ and $\kappa = \max_{\{n,b\}} \kappa_{n,b}$.*

If the class-imbalance in client selection is reduced, $\mathbb{E}[QCID]$ will decrease. Consequently, $\frac{1}{V}$ and $\frac{\mathbb{E}[QCID]}{V}$ will also decrease, making the convergence bound on the right side tighter[2]. Therefore, Theorem 4.5 not only provides a convergence guarantee for Fed-CBS, but also proves the class-imbalance reduction in client selection could benefit FL, *i.e.*, more class-balance leads to faster convergence.

## 5. Experiments

We conduct thorough experiments on three public benchmark datasets, CIFAR-10 (Krizhevsky et al.), Fashion-MNIST (Xiao et al., 2017) and FEMNIST in the Leaf Benchmark (Caldas et al., 2018). In all the experiments, we simulate cross-device federated learning (CDFL), where the system runs with a large number of clients with only a fraction of them available in each communication round, and we make client selections on those available clients. The results show that our method can achieve faster and more stable convergence compared with four baselines: random selection (rand), Power-of-choice Selection Strategy (pow-d) (Cho et al., 2020), the method in Yang et al. (2020) (Fed-cucb), and the ideal setting where we select all the available clients (all). To compare them efficiently in the main text, we present the results from Cifar-10 where the whole dataset is divided to 200 (or 120) clients, since we need to engage all the clients for the ideal setting. To simulate more realistic settings where there are thousands of clients, we conduct our method on FEMNIST in the Leaf Benchmark with more then 3000 clients. Due to the space limit, we move the results of FEMNIST, Fashion-MNIST, and the ablation studies to Section C.5 & D in the Appendix. For Fashion-MNIST, we adopt FedNova (Wang

[2]Theorem 4.5 requires the $\beta_M$ in our method to be large enough to make $\mathbb{E}[QCID] < \frac{1}{3\delta B}$ according to Theorem 3.3. How to explicitly derive a lower bound for $\beta_M$ is also very interesting and we leave it as a theoretical future work.

et al., 2020a) to show that our method can be organically integrated with existing orthogonal works which aim at improving FL.

**Experiment Setup** We target cross-device settings where the devices are resource-constrained, i.e., most of the devices do not have sufficient computational power and memory to support the training of large models. Therefore, we adopt a compact model with two convolutional layers followed by three fully-connected layers and FedAvg (McMahan et al., 2017b) as the FL optimizer. The batch size is 50 for each client. In each communication round, all of them conduct the same number of local updates, which allows the client with the largest local dataset to conduct 5 local training epochs. In our method, we set the $\beta_m = m$, $\gamma = 10$ and $L_b = 10^{-20}$. The local optimizer is SGD with a weight decay of 0.0005. The learning rate is 0.01 initially and the decay factor is 0.9992. We terminate the FL training after 3000 communication rounds and then evaluate the model's performance on the test dataset of CIFAR-10. More details of the experiment setup are listed in Section C.2.

### 5.1. Results for Class-Balanced Global Datasets

In this experiment, we set 200 clients in total with a class-balanced global dataset. The non-IID data partition among clients is based on a Dirichlet distribution parameterized by the concentration parameter $\alpha$ in Hsu et al. (2019). Roughly speaking, as $\alpha$ decreases, the data distribution will become more non-iid. In each communication round, we uniformly and randomly set 30% of them (i.e., 60 clients) available and select 10 clients from those 60 available ones to participate in the training.

As shown in Table 4, our method can achieve the lowest $QCID$ value compared with other client selection strategies. As a benefit of successfully reducing the class-imbalance, our method outperforms the other three baseline methods and achieves comparable performance to the ideal setting where all the available clients are engaged in training. As shown in Table 4 and Figure 4, our method can achieve faster and more stable convergence. The enhancement in stability can also be perceived as a reduction in gradient variance, a concept that has been explored in previous studies (Johnson & Zhang, 2013; Zhang et al., 2020b; Defazio et al., 2014; Zhao et al., 2018; Chatterji et al., 2018). It is also worth noting that due to the inaccurate distribution estimation and the limitations of the greedy method discussed in Section 2.2, the performance of Fed-cucb is much worse than ours.

### 5.2. Results for Class-Imbalanced Global Datasets

In real-world settings, the global dataset of all the clients is not always class-balanced. Hence, we investigate two different cases to show the superiority of our method and provide more details of their settings in Section C.3. To simplify the construction of a class-imbalanced global dataset, each client only has one class of data with the same quantity. We report the best test accuracy in Table 2 and present the corresponding $QCID$ values in Section C.4.

#### 5.2.1. Case 1: Uniform Availability

**Settings.** There are 120 clients in total, and the global dataset of these 120 clients is class-imbalanced. To measure the degree of class imbalance, we let the global dataset have the same amount of $n_1$ data samples for five classes and the same amount of $n_2$ data samples for the other five classes. The ratio $r$ between $n_1$ and $n_2$ is respectively set to $3:1$ and $5:1$. In each communication round, we uniformly set 30% of them (*i.e.*, 36 clients) available with replacement and select 10 clients to participate in the training.

As shown in Table 2 and Figure 5, our method can achieve faster and more stable convergence, and it even achieves slightly better performance than the ideal setting where all the available clients are engaged. The performance of Fed-cucb (Yang et al., 2020) is better than the results on the class-balanced global dataset, which is partly due to the simplicity of each client's local dataset composition in our experiments. The third line in Figure 2 indicates Fed-cucb can accurately estimate this simple type of label distribution.

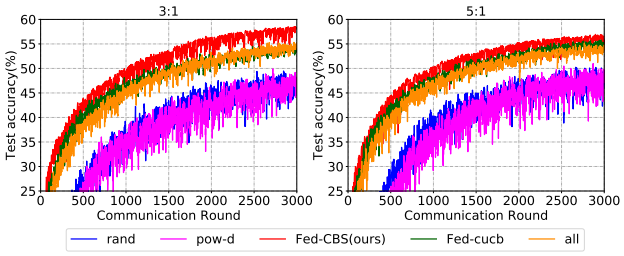



*Figure 5.* Test accuracy on Cifar-10 with class-imbalanced global dataset in Case 1.

#### 5.2.2. Case 2: Non-uniform Availability

**Settings.** There are 200 clients in total. In each communication round, 30% of them (*i.e.*, 60 clients) are set available uniformly in each training round with replacement. By non-uniformly setting the availability, the global dataset of those 60 available clients is always class-imbalanced. To measure the degree of class imbalance, we make the global dataset have the same amount of $n_1$ data samples for the five classes and have the same amount of $n_2$ data samples for the other five classes. The ratio $r$ between $n_1$ and $n_2$ is set to $3:1$ and $5:1$. We select 10 clients to participate in the training.

As shown in Table 2 and Figure 5, our method consistently achieves higher test accuracy and more stable convergence, and it also outperforms the ideal setting where all the available clients are engaged. Since the global dataset of the available 60 clients in each communication round is always class-imbalanced, engaging all of them is not the optimal selection strategy in terms of test accuracy.

| | | all | rand | pow-d | Fed-cucb | Fed-CBS |
|---|---|---|---|---|---|---|
| Case 1 | 3:1 | 55.17±0.94 | 50.99±0.97 | 53.51±0.34 | 55.11±0.26 | **56.86±0.34** |
| | 5:1 | 50.93±1.64 | 47.36±2.34 | 52.73±1.85 | 53.75±0.58 | **54.94±0.73** |
| Case 2 | 3:1 | 54.01±0.60 | 50.81±2.03 | 53.98±1.87 | 54.48±1.31 | **57.71±0.50** |
| | 5:1 | 50.42±1.27 | 48.33±3.03 | 53.54±1.18 | 53.38±1.48 | **57.99±0.46** |

*Table 2.* Best test accuracy for our method and other four baselines.

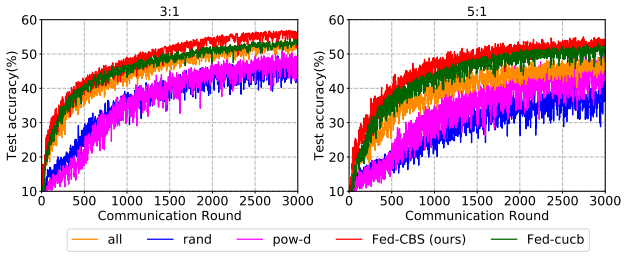



*Figure 6.* Test accuracy on Cifar-10 with class-imbalanced global dataset in Case 2.

## 6. Conclusion

We unveil the essential reason for performance degradation on non-IID data with random client selection strategy in FL training, *i.e.*, the class-imbalance. Motivated by this insight, we propose an efficient heterogeneity-aware client sampling mechanism, Fed-CBS. Extensive experiments validate that Fed-CBS significantly outperforms the status quo approaches and yields comparable or even better performance than the ideal setting where all the available clients participate in the training. We also provide the theoretical convergence guarantee of Fed-CBS. Our mechanism has numerous potential applications, including medical classification tasks. In addition, since Fed-CBS is orthogonal to most existing work to improve FL on non-IID data, it can be integrated with them to further improve the performance.

**Acknowledgments** This work is supported in part by the grants CNS-2112562, IIS-2140247, CNS-1822085 and IIS-2223292. We thank Eric Yeats, Shihan Lin and Taoan Huang for the valuable discussion and thank all reviewers for their valuable comments.

## References

Anand, R., Mehrotra, K. G., Mohan, C. K., and Ranka, S. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969, 1993.

Anati, I., Gueron, S., Johnson, S., and Scarlata, V. Innovative technology for cpu based attestation and sealing. In *Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy*, volume 13, pp. 7. Citeseer, 2013.

Balakrishnan, R., Li, T., Zhou, T., Himayat, N., Smith, V., and Bilmes, J. Diverse client selection for federated learning: Submodularity and convergence analysis. In *ICML 2021 International Workshop on Federated Learning for User Privacy and Data Confidentiality*, Virtual, July 2021.

Brakerski, Z., Gentry, C., and Vaikuntanathan, V. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

Caldas, S., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018. URL https://arxiv.org/abs/1812.01097.

Chatterji, N. S., Flammarion, N., Ma, Y.-A., Bartlett, P. L., and Jordan, M. I. On the theory of variance reduction for stochastic gradient monte carlo, 2018.

Chen, H., Laine, K., and Player, R. Simple encrypted arithmetic library-seal v2.1. In *International Conference on Financial Cryptography and Data Security*, pp. 3–18. Springer, 2017.

Chen, W., Wang, Y., and Yuan, Y. Combinatorial multi-armed bandit: General framework and applications. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 151–159, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/chen13a.html.

Chen, W., Bhardwaj, K., and Marculescu, R. Fedmax: mitigating activation divergence for accurate and communication-efficient federated learning. *arXiv preprint arXiv:2004.03657*, 2020.

Cho, Y. J., Wang, J., and Joshi, G. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *ArXiv*, abs/2010.01243, 2020.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Duan, M., Liu, D., Chen, X., Tan, Y., Ren, J., Qiao, L., and Liang, L. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In *2019 IEEE 37th international conference on computer design (ICCD)*, pp. 246–254. IEEE, 2019.

Fan, J. and Vercauteren, F. Somewhat practical fully homomorphic encryption. *IACR Cryptol. ePrint Arch.*, 2012: 144, 2012.

Fraboni, Y., Vidal, R., Kameni, L., and Lorenzi, M. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In *International Conference on Machine Learning*, pp. 3407–3416. PMLR, 2021.

Goetz, J., Malik, K., Bui, D., Moon, S., Liu, H., and Kumar, A. Active federated learning. *arXiv preprint arXiv:1909.12641*, 2019.

Halevi, S. and Shoup, V. Algorithms in HElib. In *Annual Cryptology Conference*, pp. 554–571. Springer, 2014.

Halevi, S. and Shoup, V. Bootstrapping for HElib. In *Annual International conference on the theory and applications of cryptographic techniques*, pp. 641–670. Springer, 2015.

Hao, W., El-Khamy, M., Lee, J., Zhang, J., Liang, K. J., Chen, C., and Duke, L. C. Towards fair federated learning with zero-shot data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 3310–3319, June 2021.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *ArXiv*, abs/1909.06335, 2019.

Huang, C., Li, Y., Loy, C. C., and Tang, X. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5375–5384, 2016.

Huang, T., Lin, W., Wu, W., He, L., Li, K., and Zomaya, A. Y. An efficiency-boosting client selection scheme for federated learning with fairness guarantee. *IEEE Transactions on Parallel & Distributed Systems*, 32(07): 1552–1564, jul 2021. ISSN 1558-2183. doi: 10.1109/TPDS.2020.3040887.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. Scaffold: Stochastic controlled averaging for on-device federated learning. 2019.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm, 2019.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017a.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. Communication-efficient Learning of Deep Networks from Decentralized Data. *Artificial Intelligence and Statistics*, 2017b.

Nishio, T. and Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–7, 2019.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive Federated Optimization. *arXiv e-prints*, art. arXiv:2003.00295, February 2020.

Ribero, M. and Vikalo, H. Communication-efficient federated learning via optimal client sampling. *arXiv preprint arXiv:2007.15197*, 2020.

Schneider, H. and Barker, G. *Matrices and Linear Algebra*. Dover Books on Mathematics Series. Dover Publications, 1989. ISBN 9780486660141. URL https://books.google.com/books?id=HJIT3CSb0wIC.

Shen, Z., Cervino, J., Hassani, H., and Ribeiro, A. An agnostic approach to federated learning with class imbalance. In *International Conference on Learning Representations*, 2022.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*, 2020a.

Wang, L., Xu, S., Wang, X., and Zhu, Q. Addressing class imbalance in federated learning. *arXiv preprint arXiv:2008.06217*, 2020b.

Wang, L., Xu, S., Wang, X., and Zhu, Q. Addressing class imbalance in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10165–10173, 2021.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL http://arxiv.org/abs/1708.07747. cite arxiv:1708.07747Comment: Dataset is freely available at https://github.com/zalandoresearch/fashion-mnist Benchmark is available at http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/.

Yang, M., Wong, A., Zhu, H., Wang, H., and Qian, H. Federated learning with class imbalance reduction, 2020.

Yang, Q., Zhang, J., Hao, W., Spell, G. P., and Carin, L. Flop: Federated learning on medical datasets using partial networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3845–3853, 2021.

Zhang, J., Zhang, R., Carin, L., and Chen, C. Stochastic particle-optimization sampling and the non-asymptotic convergence theory. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1877–1887. PMLR, 26–28 Aug 2020a. URL https://proceedings.mlr.press/v108/zhang20d.html.

Zhang, J., Zhao, Y., and Chen, C. Variance reduction in stochastic particle-optimization sampling. In *International Conference on Machine Learning*, pp. 11307–11316. PMLR, 2020b.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019.

Zhao, Y., Zhang, J., and Chen, C. Self-adversarially learned bayesian sampling, 2018.

# A. Privacy Protection in the framework

## A.1. Proof of Theorem 3.2

*Proof.* By the definitions of $\{\boldsymbol{\alpha_i}\}$, we define the following matrix $\boldsymbol{A}_\alpha$

$$\boldsymbol{A}_\alpha \triangleq \begin{bmatrix} q_1\boldsymbol{\alpha}_1 \\ \cdots \\ q_n\boldsymbol{\alpha}_n \\ \cdots \\ q_N\boldsymbol{\alpha}_N \end{bmatrix} = \begin{bmatrix} q_1\alpha_{(1,1)} & q_1\alpha_{(1,2)} & \cdots & q_1\alpha_{(1,b)} & \cdots & q_1\alpha_{(1,B)} \\ \cdots \\ q_n\alpha_{(n,1)} & q_n\alpha_{(n,2)} & \cdots & q_n\alpha_{(n,b)} & \cdots & q_n\alpha_{(n,B)} \\ \cdots \\ q_N\alpha_{(N,1)} & q_N\alpha_{(N,2)} & \cdots & q_N\alpha_{(N,b)} & \cdots & q_N\alpha_{(N,B)} \end{bmatrix}$$

By the definitions of $\boldsymbol{S}$, we have

$$\boldsymbol{S} = \boldsymbol{A}_\alpha \cdot \boldsymbol{A}_\alpha^\intercal \tag{2}$$

To derive the exact values of $\{\boldsymbol{\alpha_i}\}$ based on $\boldsymbol{S}$, we need to solve the problem 2. However, given $\boldsymbol{S}$, the $\boldsymbol{A}_\alpha$ which satisfies $\boldsymbol{S} = \boldsymbol{A}_\alpha \cdot \boldsymbol{A}_\alpha^\intercal$ is not unique. If $\bar{\boldsymbol{A}}_\alpha$ is a solution to the problem 2, then for any orthogonal matrix $\boldsymbol{Q}$ *i.e.* $\boldsymbol{Q} \cdot \boldsymbol{Q}^\intercal = \boldsymbol{I}$ where the $\boldsymbol{I}$ is the identity matrix, the new matrix $\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{Q}$ is also solution to the problem 2. This is because

$$\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{Q} \cdot (\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{Q})^\intercal = \bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{Q} \cdot \boldsymbol{Q}^\intercal \cdot \bar{\boldsymbol{A}}_\alpha^\intercal = \bar{\boldsymbol{A}}_\alpha \cdot \bar{\boldsymbol{A}}_\alpha^\intercal = \boldsymbol{S}$$

Hence, the $\boldsymbol{A}_\alpha$ which satisfies $\boldsymbol{S} = \boldsymbol{A}_\alpha \cdot \boldsymbol{A}_\alpha^\intercal$ is not unique and we finish our proof. □

To understand the Theorem 3.2, we provide the following example. We can conduct the following permutation on the columns of $\boldsymbol{A}_\alpha$ (*i.e.* moving the first column to the place before the last column), we can derive a new matrix $\bar{\boldsymbol{A}}_\alpha$.

$$\bar{\boldsymbol{A}}_\alpha \triangleq \begin{bmatrix} q_1\alpha_{(1,2)} & \cdots & q_1\alpha_{(1,b)} & \cdots & q_1\alpha_{(1,1)} & q_1\alpha_{(1,B)} \\ \cdots \\ q_n\alpha_{(n,2)} & \cdots & q_n\alpha_{(n,b)} & \cdots & q_n\alpha_{(n,1)} & q_n\alpha_{(n,B)} \\ \cdots \\ q_N\alpha_{(N,2)} & \cdots & q_N\alpha_{(N,b)} & \cdots & q_N\alpha_{(N,1)} & q_N\alpha_{(N,B)} \end{bmatrix}$$

We can find that $\bar{\boldsymbol{A}}_\alpha$ also satisfies $\boldsymbol{S} = \bar{\boldsymbol{A}}_\alpha \cdot \bar{\boldsymbol{A}}_\alpha^\intercal$. Actually, there are also many other permutations that can derive the solutions to the problem 2. Hence, in our framework shown in 7, the selector can not estimate the exact label distribution of the clients.

## A.2. An Example of Deriving $S$ Using FHE

FHE (Brakerski et al., 2014; Fan & Vercauteren, 2012; Halevi & Shoup, 2015) enables an untrusted party to perform computation (addition and multiplication) on encrypted data. In Figure 7, we provide a framework as an example to show it is possible to derive $S$ without knowing the values of local label distributions $\{\boldsymbol{\alpha_i}\}$ using FHE. Our framework can be realized using off-the-shelf FHE libraries such as (Chen et al., 2017).

There is a selector in our example. It is usually from a third party and keeps a unique private key, denoted by $K_1^{-1}$. The corresponding public key is denoted by $K_1$. In the confidential transmission between server and clients, each client first uses $K_1$ to encrypt their label distribution vector $\boldsymbol{\alpha_k}$ as $K_1(\boldsymbol{\alpha_k})$, and transmits it to the server. Since only the server has access to $K_1(\boldsymbol{\alpha_k})$, no one else including the selector can decrypt it and get $\boldsymbol{\alpha_k}$. When the server gets all $K_1(\boldsymbol{\alpha_k})$, it will conduct FHE computation to get the matrix $K_1(S) = K_1(\{\boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j\}_{ij}) = \{K_1(\boldsymbol{\alpha}_i)^T K_1(\boldsymbol{\alpha}_j)\}_{ij}$. Then the server transmits the $K_1(S)$ to selector, and selector uses $K_1^{-1}$ to access the final result $S$. Since only the selector has $K_1^{-1}$, only it knows $S$. After that, the selector will conduct client selection following some strategy to derive the result $\mathcal{M}$ and transmit it back to the server. At last, the server will collect the model parameters of the clients in $\mathcal{M}$ and conduct FL aggregation. In the whole process, the server, selector or any other clients except client $n$ can not get $\alpha_n$. Furthermore, all clients and servers have no access to the inner product results $S$, which prevents malicious clients or servers from inferring the label distributions of the other clients.

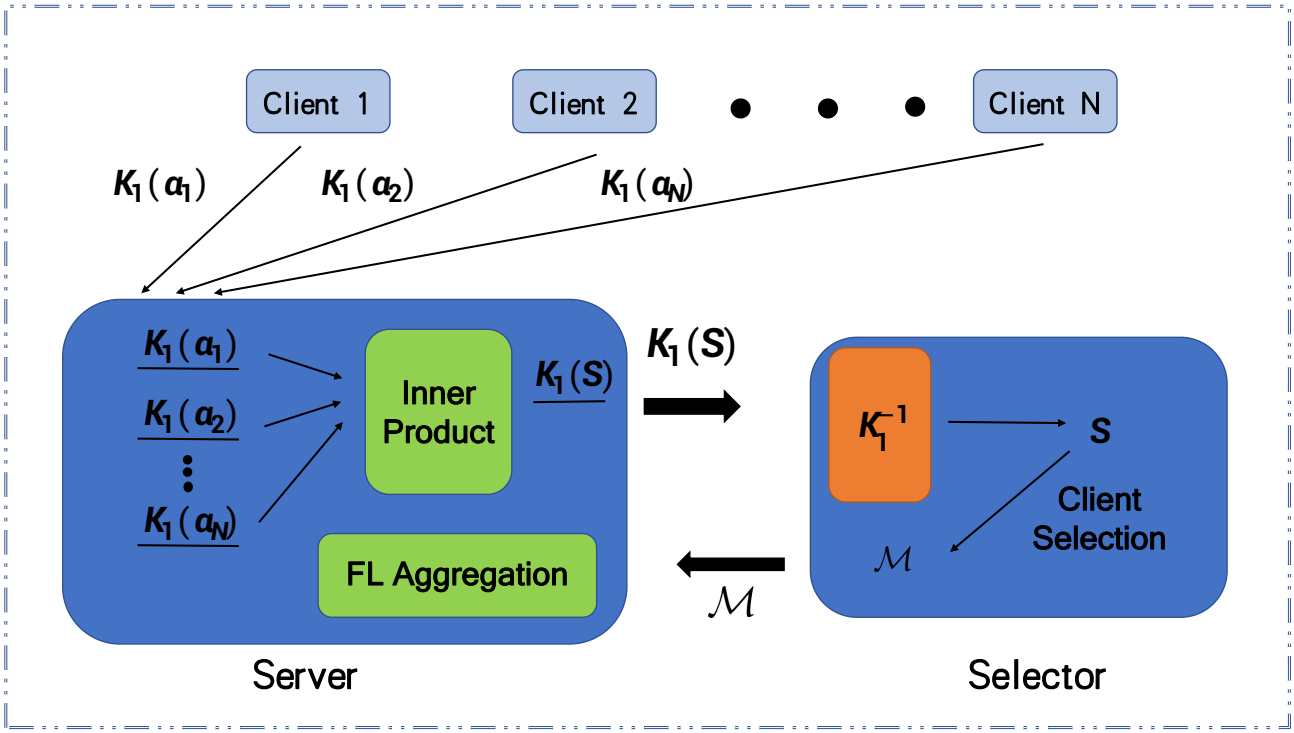



*Figure 7.* An example of FHE to securely transmit $S$.

The server, selector or any other clients except client $n$ can not get $\alpha_n$, which protects the privacy of the clients. Furthermore, only the clients have no access to the inner product results $S$, which prevents malicious clients or servers from inferring the label distributions of the other clients. We also prove that it is impossible even for the selector to derive $\{\boldsymbol{\alpha_i}\}$ from $S$ with theorem 3.2.

## B. Proof of Theorem 3.1, 3.3, 3.4 and 4.5

### B.1. Proof of Theorem 3.1

*Proof.*

$$
\begin{aligned}
QCID(\mathcal{M}) &= \sum_{b=1}^{B}(\frac{\sum_{n\in\mathcal{M}} q_n\alpha_{(n,b)}}{\sum_{n\in\mathcal{M}} q_n} - \frac{1}{B})^2 \\
&= \sum_{b=1}^{B}(\frac{\sum_{n\in\mathcal{M}} q_n\alpha_{(n,b)}}{\sum_{n\in\mathcal{M}} q_n})^2 - 2 * \frac{1}{\sum_{n\in\mathcal{M}} q_n} * \frac{1}{B} * \sum_{n\in\mathcal{M}} q_n + B * \frac{1}{B^2} \\
&= \sum_{b=1}^{B}(\frac{\sum_{n\in\mathcal{M}} q_n\alpha_{(n,b)}}{\sum_{n\in\mathcal{M}} q_n})^2 - \frac{1}{B} \\
&= \frac{1}{(\sum_{n\in\mathcal{M}} q_n)^2} \sum_{n\in\mathcal{M},n'\in\mathcal{M}} q_n q_n' (\sum_{b=1}^{B} \alpha_{(n,b)}\alpha_{(n',b)}) - \frac{1}{B} \\
&= \frac{1}{(\sum_{n\in\mathcal{M}} q_n)^2} \sum_{n\in\mathcal{M},n'\in\mathcal{M}} q_n q_n' \boldsymbol{\alpha}_n \boldsymbol{\alpha}_{n'}^T - \frac{1}{B}
\end{aligned}
$$

□

### B.2. Proof of Theorem 3.3

*Proof.* To select $M$ clients from $N$ available clients, there are $\binom{N}{M}$ different choices to construct $\mathcal{M}$, denoted by $\mathcal{M}^{(1)}, \mathcal{M}^{(2)}, ..., \mathcal{M}^{(\binom{N}{M})}$, respectively. Let $x_i \triangleq QCID(\mathcal{M}^{(i)})$ and $\bar{N} \triangleq (\binom{N}{M})$. Then we have

$$\mathbb{E}_{\mathcal{M}\sim P_{\beta_M}} QCID(\mathcal{M}) = x_1 \frac{\frac{1}{x_1^{\beta_M}}}{\frac{1}{x_1^{\beta_M}} + \frac{1}{x_2^{\beta_M}} + ... + \frac{1}{x_{\bar{N}}^{\beta_M}}} + x_2 \frac{\frac{1}{x_2^{\beta_M}}}{\frac{1}{x_1^{\beta_M}} + \frac{1}{x_2^{\beta_M}} + ... + \frac{1}{x_{\bar{N}}^{\beta_M}}} + ... + x_{\bar{N}} \frac{\frac{1}{x_{\bar{N}}^{\beta_M}}}{\frac{1}{x_1^{\beta_M}} + \frac{1}{x_2^{\beta_M}} + ... + \frac{1}{x_{\bar{N}}^{\beta_M}}}$$

$$\text{And } \mathbb{E}_{\mathcal{M}\sim P_{rand}} QCID(\mathcal{M}) = \frac{1}{\bar{N}}(x_1 + x_2 + ... + x_{\bar{N}})$$

Without loss of generality, we assume $x_1 \leq x_2 \leq ... \leq x_{\bar{N}}$ and define the following $y_i$ for the notation simplicity:

$$y_i = \begin{cases} \frac{1}{x_i^{\beta_M}} & \text{if} \quad 0 \leq i \leq \bar{N} \\ \frac{1}{x_{i-\bar{N}}^{\beta_M}} & \text{if} \quad \bar{N} < i \leq 2\bar{N} - 1 \end{cases} \tag{3}$$

Now we calculate the following ratio:

$$\begin{aligned} \frac{\mathbb{E}_{\mathcal{M}\sim P_{\beta_M}} QCID(\mathcal{M})}{\mathbb{E}_{\mathcal{M}\sim P_{rand}} QCID(\mathcal{M})} &= \frac{\bar{N}(x_1 \frac{1}{x_1^{\beta_M}} + x_2 \frac{1}{x_2^{\beta_M}} + ... + x_{\bar{N}} \frac{1}{x_{\bar{N}}^{\beta_M}})}{(x_1 + x_2 + ... + x_{\bar{N}})(\frac{1}{x_1^{\beta_M}} + \frac{1}{x_2^{\beta_M}} + ... + \frac{1}{x_{\bar{N}}^{\beta_M}})} \\ &= \frac{\sum_{j=1}^{\bar{N}} (\sum_{i=1}^{\bar{N}} x_i \frac{1}{x_i^{\beta_M}})}{\sum_{j=1}^{\bar{N}} (\sum_{i=1}^{\bar{N}} x_i y_{j+i-1})} \end{aligned}$$

Since we assume that $x_1 \leq x_2 \leq ... \leq x_{\bar{N}}$, we have $\frac{1}{x_1^{\beta_M}} \geq \frac{1}{x_2^{\beta_M}} \geq ... \geq \frac{1}{x_{\bar{N}}^{\beta_M}}$. Besides, it is easy to find $x_i$ and $x_{i'}$ satisfying $x_i \neq x_{i'}$. Then for each $1 \leq j \leq \bar{N}$, according to the rearrangement inequality, we have

$$\begin{aligned} &\sum_{i=1}^{\bar{N}} x_i \frac{1}{x_i^{\beta_M}} < \sum_{i=1}^{\bar{N}} x_i y_{j+i-1} \Rightarrow \sum_{j=1}^{\bar{N}} \sum_{i=1}^{\bar{N}} x_i \frac{1}{x_i^{\beta_M}} < \sum_{j=1}^{\bar{N}} \sum_{i=1}^{\bar{N}} x_i y_{j+i-1} \Rightarrow \\ &\frac{\mathbb{E}_{\mathcal{M}\sim P_{\beta_M}} QCID(\mathcal{M})}{\mathbb{E}_{\mathcal{M}\sim P_{rand}} QCID(\mathcal{M})} < 1 \Rightarrow \mathbb{E}_{\mathcal{M}\sim P_{\beta_M}} QCID(\mathcal{M}) < \mathbb{E}_{\mathcal{M}\sim P_{rand}} QCID(\mathcal{M}) \end{aligned}$$

Similarly, for $\beta'_M$ such that $\beta'_M \geq \beta_M$, denote $\beta'_M = \beta_M + \Delta\beta$. We have

$$\begin{aligned} &\mathbb{E}_{\mathcal{M}\sim P_{\beta'_M}} QCID(\mathcal{M}) = \\ &x_1 \frac{\frac{1}{x_1^{\beta'_M}}}{\frac{1}{x_1^{\beta'_M}} + \frac{1}{x_2^{\beta'_M}} + ... + \frac{1}{x_{\bar{N}}^{\beta'_M}}} + x_2 \frac{\frac{1}{x_2^{\beta'_M}}}{\frac{1}{x_1^{\beta'_M}} + \frac{1}{x_2^{\beta'_M}} + ... + \frac{1}{x_{\bar{N}}^{\beta'_M}}} + ... + x_{\bar{N}} \frac{\frac{1}{x_{\bar{N}}^{\beta'_M}}}{\frac{1}{x_1^{\beta'_M}} + \frac{1}{x_2^{\beta'_M}} + ... + \frac{1}{x_{\bar{N}}^{\beta'_M}}} = \\ &\frac{x_1 \frac{1}{x_1^{\beta_M + \Delta\beta}}}{\frac{1}{x_1^{\beta_M + \Delta\beta}} + \frac{1}{x_2^{\beta_M + \Delta\beta}} + ... + \frac{1}{x_{\bar{N}}^{\beta_M + \Delta\beta}}} + \frac{x_2 \frac{1}{x_2^{\beta_M + \Delta\beta}}}{\frac{1}{x_1^{\beta_M + \Delta\beta}} + \frac{1}{x_2^{\beta_M + \Delta\beta}} + ... + \frac{1}{x_{\bar{N}}^{\beta_M + \Delta\beta}}} + ... + \frac{x_{\bar{N}} \frac{1}{x_{\bar{N}}^{\beta_M + \Delta\beta}}}{\frac{1}{x_1^{\beta_M + \Delta\beta}} + \frac{1}{x_2^{\beta_M + \Delta\beta}} + ... + \frac{1}{x_{\bar{N}}^{\beta_M + \Delta\beta}}} \end{aligned}$$

Now we calculate the following ratio:

$$\frac{\mathbb{E}_{\mathcal{M}\sim P_{\beta'_M}} QCID(\mathcal{M})}{\mathbb{E}_{\mathcal{M}\sim P_{\beta_M}} QCID(\mathcal{M})} = \frac{\left(\frac{x_1}{x_1^{\beta_M+\Delta\beta}} + \frac{x_2}{x_2^{\beta_M+\Delta\beta}} + ... + \frac{x_N}{x_N^{\beta_M+\Delta\beta}}\right)\left(\frac{1}{x_1^{\beta_M}} + \frac{1}{x_2^{\beta_M}} + ... + \frac{1}{x_{\bar{N}}^{\beta_M}}\right)}{\left(\frac{1}{x_1^{\beta_M+\Delta\beta}} + \frac{1}{x_2^{\beta_M+\Delta\beta}} + ... + \frac{1}{x_{\bar{N}}^{\beta_M+\Delta\beta}}\right)\left(\frac{x_1}{x_1^{\beta_M}} + \frac{x_2}{x_2^{\beta_M}} + ... + \frac{x_{\bar{N}}}{x_{\bar{N}}^{\beta_M}}\right)}$$
$$= \frac{\sum_{1\le i\le j\le N} \frac{1}{x_i^{\beta_M} x_j^{\beta_M}}\left(\frac{x_i}{x_i^{\Delta\beta}} + \frac{x_j}{x_j^{\Delta\beta}}\right)}{\sum_{1\le i\le j\le N} \frac{1}{x_i^{\beta_M} x_j^{\beta_M}}\left(\frac{x_i}{x_j^{\Delta\beta}} + \frac{x_j}{x_i^{\Delta\beta}}\right)}$$

Since we assume that $x_1 \le x_2 \le ... \le x_{\bar{N}}$, we have $\frac{1}{x_1^{\Delta\beta}} \ge \frac{1}{x_2^{\Delta\beta}} \ge ... \ge \frac{1}{x_{\bar{N}}^{\Delta\beta}}$. Then for each $1 \le i \le j \le N$, according to the rearrangement inequality, we have

$$\frac{x_i}{x_i^{\Delta\beta}} + \frac{x_j}{x_j^{\Delta\beta}} \le \frac{x_i}{x_j^{\Delta\beta}} + \frac{x_j}{x_i^{\Delta\beta}}$$

Furthermore, among all the $(x_i, x_j)$ pairs, it is easy to find one $(x_i, x_{i'})$ such that it satisfies $x_i \ne x_{i'}$. Thus we have

$$\frac{x_i}{x_i^{\Delta\beta}} + \frac{x_{i'}}{x_{i'}^{\Delta\beta}} < \frac{x_i}{x_{i'}^{\Delta\beta}} + \frac{x_{i'}}{x_i^{\Delta\beta}}$$

Consequently, we have

$$\sum_{1\le i\le j\le N} \frac{1}{x_i^{\beta_M} x_j^{\beta_M}}\left(\frac{x_i}{x_i^{\Delta\beta}} + \frac{x_j}{x_j^{\Delta\beta}}\right) < \sum_{1\le i\le j\le N} \frac{1}{x_i^{\beta_M} x_j^{\beta_M}}\left(\frac{x_i}{x_j^{\Delta\beta}} + \frac{x_j}{x_i^{\Delta\beta}}\right) \Rightarrow$$
$$\frac{\mathbb{E}_{\mathcal{M}\sim P_{\beta'_M}} QCID(\mathcal{M})}{\mathbb{E}_{\mathcal{M}\sim P_{\beta_M}} QCID(\mathcal{M})} < 1 \Rightarrow \mathbb{E}_{\mathcal{M}\sim P_{\beta'_M}} QCID(\mathcal{M}) < \mathbb{E}_{\mathcal{M}\sim P_{\beta_M}} QCID(\mathcal{M})$$

□

### B.3. Proof of NP-hardness

We provide the following proof to prove the NP-hardness. First, we need to clarify the definitions of the following three problems.

**Problem 1**: We need to select M clients among N clients such that the grouped dataset of these M clients is class-balanced. There are B (B 2) classes in total. Our goal is to prove the NP-hardness of Problem 1.

**Problem 2**: We need to select N clients among 2 N clients such that the group dataset of these N clients is class-balanced. There are 2 classes in total. We denote the distribution of the local dataset of the $n$-th client as $[x_n, y_n]$, where $x_n$ and $y_n$ are non-negative integers.

Problem 2 is a particular case of Problem 1. If we can prove the NP-hardness of Problem 2, then Problem 1 is also NP-hard.

**Problem 3 (Partition problem)**: Deciding whether a given multiset $S$ of $K$ positive integers can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ equals the sum of the numbers in $S_1$. We denote the $S$ as $\{s_1, s_2, \ldots, s_K\}$

It is well-known that the Partition problem is an NP-complete problem. Hence the overall idea of our proof is to reduce Problem 2 to Problem 3. Then we can show that Problem 2 is NP-hard.

*Proof.* Case 1: We first consider the case where $K$ is an even number, where $K = 2N$. We denote the sum of all the elements in $S$ as $W$, where $W = s_1 + s_2 + \ldots + s_K$. We define a new positive value $P$ as

$$P = \min\{\,|2W - Ks_1|\,, |2W - Ks_2|\,, \ldots, |2W - Ks_K|\} + 1$$

Now, we define the following non-negative $x_n$ and $y_n$, where $1 \leq n \leq K$

$$x_1 = Ks_1 + P, \quad y_1 = 2W - Ks_1 + P$$
$$x_2 = Ks_2 + P, \quad y_2 = 2W - Ks_2 + P$$
$$x_K = Ks_K + P, \quad y_K = 2W - Ks_K + P$$

Now we can consider the $[x_1, y_1], [x_2, y_2], \ldots, [x_K, y_K]$ as the class distributions defined in Problem 2 . If Problem 2 is not NP-hard, we can find $N = \frac{K}{2}$ clients among the above $K$ clients such that the grouped dataset is class-balanced within polynomial time complexity. We denote those $N$ clients' distribution as $[\bar{x}_1, \bar{y}_1], [\bar{x}_2, \bar{y}_2], \ldots, [\bar{x}_K, \bar{y}_N]$. Then we denote the corresponding elements in $S$ as $\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_N$. Since it is class-balanced solution, we have

$$\bar{x}_1 + \bar{x}_2 + \ldots + \bar{x}_N = \bar{y}_1 + \bar{y}_2 + \ldots + \bar{y}_N$$

By summarizing all the $\bar{x}_n$ and $\bar{y}_n$. we can derive that $(\bar{x}_1 + \bar{y}_1) + (\bar{x}_2 + \bar{y}_2) + \ldots + (\bar{x}_N + \bar{y}_N) = N(2W + 2P)$. Then we have $\bar{x}_1 + \bar{x}_2 + \ldots + \bar{x}_N = N(W + P)$ According to the definition of $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N$, we have

$$(K\bar{s}_1 + P) + (K\bar{s}_2 + P) + \ldots + (K\bar{s}_N + P) = N(W + P)$$

Since $K = 2N$ we have $\bar{s}_1 + \bar{s}_2 + \ldots + \bar{s}_N = \frac{W}{2}$. This means we can solve the Partition problem within polynomial time complexity when $K$ is an even number. Case 2: If $K$ is an odd number, where $K = 2N - 1$, we can just add an auxiliary element $s\prime = 0$ to the original $S$ and derive a new set $St = S \cup \{s\prime\}$. If Problem 2 is not NP-hard, we can follow the same process as in Case 1 to solve the Partition problem within polynomial time complexity when K is an odd number.

We know these solutions to Case 2& Case 1 conflict with the fact that the Partition problem is NP-hard. Hence, Problem 2 is NP-hard. Then Problem 1 is NP-hard, and we finish our proof.

□

### B.4. Proof of Theorem 3.4

*Proof.* According to (Schneider & Barker, 1989), we first define the principle submatrix, which is a submatrix where the set of remaining row indices is the same as the remaining set of column indices .

Before selecting the first client, we need to calculate the following value for all clients $c_1 \in \{1, 2, 3, ..., N\}$,

$$P(C_1 = c_1) \propto \frac{1}{[QCID(\mathcal{M}_1)]^{\beta_1}} + \lambda\sqrt{\frac{3\ln k}{2T_{c_1}}}, \quad \beta_1 > 0.$$

To derive the $QCID(\mathcal{M}_1)$ for each $c_1 \in \{1, 2, 3, ..., N\}$, according to Theorem 3.1, we need to find the principle submatrix of S, denoted by $S_1$, in which the set of column indices is $\mathcal{M}_1$. Then we need to calculate the sum of all the elements in $S_1$. Since there are $N$ different values for $c_1$ and the dimension of $S_1$ is $1 \times 1$, we need to conduct the computation for $N$ times.

After selecting $\mathcal{M}_1 = \{c_1\}$, we need to select $c_2 \in \{1, 2, 3, ..., N\}/\mathcal{M}_1$ to form $\mathcal{M}_2 = \mathcal{M}_1 \bigcup \{c_2\}$.

Before selecting the second client, we need to calculate the following value for all the $\mathcal{M}_2 = \{c_1, c_2\}$ where $c_2 \in \{1, 2, 3, ..., N\}/\mathcal{M}_1$,

$$P(C_2 = c_2 | C_1 = c_1) \propto \frac{\frac{1}{[QCID(\mathcal{M}_2)]^{\beta_2}}}{\frac{1}{[QCID(\mathcal{M}_1)]^{\beta_1}} + \alpha\sqrt{\frac{3\ln k}{2T_{c_1}}}}$$

To derive the $QCID(\mathcal{M}_2)$ for each $c_2 \in \{1, 2, 3, ..., N\}/\{\mathcal{M}_1\}$, according to Theorem 3.1, we need to find the principle submatrix of S, denoted by $S_2$, in which the set of column indices is $\mathcal{M}_2$. Then we need to calculate the sum of all the elements in $S_2$. Since there are $N - 1$ different values for $c_2$, there will be $N - 1$ different $S_2$. Also, because we have already calculate the sum of all the elements in $S_1$, which is a submatrix of $S_2$, in our first step, we now only need to sum over all the other elements in $S_2$. Since the dimension of $S_2$ is $2 \times 2$, we need to do the computation for $(N - 1) \times (2^2 - 1)$ times.

This procedure goes on. After selecting $\mathcal{M}_{m-1} = \{c_1, c_2, ..., c_{m-1}\}$, where $3 \leq m \leq M$, we need to select $c_m \in \{1, 2, 3, ..., N\}/\mathcal{M}_m$ to form $\mathcal{M}_m = \mathcal{M}_{m-1} \bigcup \{c_m\}$. Before selecting the $m$-th client, we need to calculate the following value for all the $\mathcal{M}_m = \{c_1, c_2, ..., c_m\}$ where $c_m \in \{1, 2, 3, ..., N\}/\mathcal{M}_{m-1}$,

$$P(C_m = c_m | C_{m-1} = c_{m-1}, ..., C_2 = c_2, C_1 = c_1) \propto \frac{[QCID(\mathcal{M}_{m-1})]^{\beta_{m-1}}}{[QCID(\mathcal{M}_m)]^{\beta_m}}$$

To derive the $QCID(\mathcal{M}_m)$ for each $c_m \in \{1, 2, 3, ..., N\}/\{\mathcal{M}_{m-1}\}$, according to Theorem 3.1, we need to find the principle submatrix of S, denoted by $S_m$, in which the set of column indices is $\mathcal{M}_m$. Then we need to calculate the sum of all the elements in $S_m$. Since there are $N - (m-1)$ different values for $c_m$, there will be $N - (m-1)$ different $S_m$. Since we have already calculate the sum of all the elements in $S_{m-1}$, which is a submatrix of $S_m$, in our previous step, now we only need to sum all the other elements in $S_m$. Since the dimension of $S_m$ is $m \times m$, we need to conduct the computation for $(N - (m-1)) \times (m^2 - (m-1)^2)$ times.

In summary, in our strategy, the total times of computations we need to conduct are

$$\begin{aligned}
&N + (N-1) \times (2^2 - 1) + ... + (N - (m-1)) \times (m^2 - (m-1)^2) + ... + (N - M) \times (M^2 - (M-1)^2) \\
&\leq N + N \times (2^2 - 1) + ... + N \times (m^2 - (m-1)^2) + ... + N \times (M^2 - (M-1)^2) \\
&= N \times M^2 \,,
\end{aligned}$$

which finishes the proof that the computation complexity for our method is $\mathcal{O}\left(N \times M^2\right)$. □

### B.5. Proof of Theorem 4.5

*Proof.* Suppose there are N available clients and their indices are denoted by $\{1, 2, 3, .., N\}$. Our goal is to get a subset $\mathcal{M}$ of $\{1, 2, 3, .., N\}$ following the probability law $S$ of some client selection strategy. Let $\boldsymbol{w}_n^{(k,t)}$ denote the model parameter of client $n$ after $t$ local updates in the $k$-th communication round and $\boldsymbol{w}^{(k,0)}$ denote the global model parameter at the beginning of the $k$-th communication round. According to the proof of Theorem 1 in (Wang et al., 2020a), we can define the following auxiliary variables for the setting where we adopt FedAvg as the FL optimizer and all the client conduct $\tau$ local updates in each communication round $k$:

Normalized Stochastic Gradient: $\boldsymbol{d}_n^{(k)} = \frac{1}{\tau} \sum_{k=0}^{\tau-1} g_n \left(\boldsymbol{w}_n^{(k,t)}\right)$,

Normalized Gradient: $\boldsymbol{h}_n^{(k)} = \frac{1}{\tau} \sum_{k=0}^{\tau-1} \nabla F_n \left(\boldsymbol{w}_n^{(k,t)}\right)$.

Normalized Class-wise Gradient: $\boldsymbol{h}_{(n,b)}^{(k)} = \frac{1}{\tau} \sum_{k=0}^{\tau-1} \nabla F_{(n,b)} \left(\boldsymbol{w}_n^{(k,t)}\right)$.

It is easy to verify that $\boldsymbol{h}_n^{(k)} = \sum_{b=1}^{B} \alpha_{(n,b)} \boldsymbol{h}_{(n,b)}^{(k)}$.

According to the proof of Theorem 1 in (Wang et al., 2020a), one can show that $\mathbb{E}\left[\boldsymbol{d}_n^{(k)} - \boldsymbol{h}_n^{(k)}\right] = 0$. Besides, since clients are independent to each other, we have $\mathbb{E}\left\langle \boldsymbol{d}_n^{(k)} - \boldsymbol{h}_n^{(k)}, \boldsymbol{d}_{n'}^{(k)} - \boldsymbol{h}_{n'}^{(k)} \right\rangle = 0, \forall n \neq n'$. Recall that the update rule of the global model can be written as follows:

$$\boldsymbol{w}^{(k+1,0)} - \boldsymbol{w}^{(k,0)} = -\eta \frac{\sum_{n \in \mathcal{M}} q_n \boldsymbol{d}_n^{(k)}}{\sum_{n \in \mathcal{M}} q_n},$$

where $\eta$ is the learning rate. According to the Lipschitz-smooth assumption for the global objective function $\widetilde{F}$ (Asssumption 4.1), it follows that

$$\begin{aligned}
&\mathbb{E}\left[\widetilde{F}\left(\boldsymbol{w}^{(k+1,0)}\right)\right] - \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right) \\
\leq &-\eta \underbrace{\mathbb{E}\left[\left\langle \nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right), \frac{\sum_{n \in \mathcal{M}} q_n \boldsymbol{d}_n^{(k)}}{\sum_{n \in \mathcal{M}} q_n} \right\rangle\right]}_{T_1} + \frac{\eta^2 L_{\widetilde{F}}}{2} \underbrace{\mathbb{E}\left[\left\| \frac{\sum_{n \in \mathcal{M}} q_n \boldsymbol{d}_n^{(k)}}{\sum_{n \in \mathcal{M}} q_n} \right\|^2\right]}_{T_2}
\end{aligned} \tag{4}$$

where the expectation is taken over randomly selected indices set $\mathcal{M}$ as well as mini-batches $\xi_i^{(k,t)}, \forall n \in \{1,2,\ldots,m\}, t \in \{0,1,\ldots,\tau-1\}$

Similar to the proof in (Wang et al., 2020a), to bound the $T_1$ in (4), we should notice that

$$\begin{aligned}
T_1 &= \mathbb{E}\left[\left\langle \nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right), \frac{\sum_{n\in\mathcal{M}} q_n\left(\boldsymbol{d}_n^{(k)} - \boldsymbol{h}_n^{(k)}\right)}{\sum_{n\in\mathcal{M}} q_n}\right\rangle\right] + \mathbb{E}\left[\left\langle \nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right), \frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\rangle\right] \\
&= \mathbb{E}\left[\left\langle \nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right), \frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\rangle\right] \\
&= \frac{1}{2}\left\|\nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2 + \frac{1}{2}\mathbb{E}\left[\left\|\frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\|^2\right] - \frac{1}{2}\mathbb{E}\left[\left\|\nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right) - \frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\|^2\right]
\end{aligned} \tag{5}$$

where the last equation uses the fact: $2\langle a, b\rangle = \|a\|^2 + \|b\|^2 - \|a-b\|^2$.

$T_2$ is similar as the one in (Wang et al., 2020a). According to the proof in Section C.3 of (Wang et al., 2020a) , we have the following bound for $T_2$,

$$\begin{aligned}
T_2 \leq & 2\sigma^2 \mathbb{E}\frac{\sum_{n\in\mathcal{M}} q_n^2}{(\sum_{n\in\mathcal{M}} q_n)^2} + 2\mathbb{E}\left[\left\|\frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\|^2\right] \\
\leq & 2\sigma^2 + 2\mathbb{E}\left[\left\|\frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\|^2\right]
\end{aligned} \tag{6}$$

Plugging (5) and (6) back into (4), we have

$$\begin{aligned}
& \mathbb{E}\left[\widetilde{F}\left(\boldsymbol{w}^{(k+1,0)}\right)\right] - \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right) \\
\leq & -\eta \underbrace{\mathbb{E}\left[\left\langle \nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right), \frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{d}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\rangle\right]}_{T_1} + \frac{\eta^2 L_{\widetilde{F}}}{2} \underbrace{\mathbb{E}\left[\left\|\frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{d}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\|^2\right]}_{T_2} \\
\leq & -\frac{1}{2}\eta\left\|\nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2 - \frac{1}{2}\eta\mathbb{E}\left[\left\|\frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\|^2\right] + \frac{1}{2}\eta\mathbb{E}\left[\left\|\nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right) - \frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\|^2\right] \\
& + \eta^2 L_{\widetilde{F}}\sigma^2 + \eta^2 L_{\widetilde{F}}\mathbb{E}\left[\left\|\frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\|^2\right]
\end{aligned} \tag{7}$$

If we set $\eta \leq \frac{1}{2L}$, we have

$$\begin{aligned}
& \mathbb{E}\left[\widetilde{F}\left(\boldsymbol{w}^{(k+1,0)}\right)\right] - \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right) \\
\leq & -\frac{1}{2}\eta\left\|\nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2 + \frac{1}{2}\eta\mathbb{E}\left[\left\|\nabla \widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right) - \frac{\sum_{n\in\mathcal{M}} q_n \boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}} q_n}\right\|^2\right] + \eta^2 L_{\widetilde{F}}\sigma^2.
\end{aligned} \tag{8}$$

Now we focus on the $\mathbb{E}\left[\left\|\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)-\frac{\sum_{n\in\mathcal{M}}q_n\boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}}q_n}\right\|^2\right]$ in the following:

$$
\begin{aligned}
&\mathbb{E}\left\|\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)-\frac{\sum_{n\in\mathcal{M}}q_n\boldsymbol{h}_n^{(k)}}{\sum_{n\in\mathcal{M}}q_n}\right\|^2=\mathbb{E}\left\|\frac{1}{B}\sum_{b=1}^{B}\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)-\frac{\sum_{n\in\mathcal{M}}q_n\left(\sum_{b=1}^{B}\alpha_{(n,b)}\boldsymbol{h}_{(n,b)}^{(k)}\right)}{\sum_{n\in\mathcal{M}}q_n}\right\|^2\\
&=\mathbb{E}\left\|\frac{1}{B}\sum_{b=1}^{B}\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)-\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}\boldsymbol{h}_{(n,b)}^{(k)}}{\sum_{n\in\mathcal{M}}q_n}\right\|^2\\
&\leq 2\mathbb{E}\left\|\frac{1}{B}\sum_{b=1}^{B}\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)-\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)}{\sum_{n\in\mathcal{M}}q_n}\right\|^2\\
&+2\mathbb{E}\left\|\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)}{\sum_{n\in\mathcal{M}}q_n}-\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}\boldsymbol{h}_{(n,b)}^{(k)}}{\sum_{n\in\mathcal{M}}q_n}\right\|^2\\
&=\underbrace{2\,\mathbb{E}\left\|\sum_{b=1}^{B}(\frac{1}{B}-\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}}{\sum_{n\in\mathcal{M}}q_n})\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2}_{T_3}+\underbrace{2\,\mathbb{E}\left\|\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}[\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)-\boldsymbol{h}_{(n,b)}^{(k)}]}{\sum_{n\in\mathcal{M}}q_n}\right\|^2}_{T_4}
\end{aligned}
\tag{9}
$$

For $T_3$, according to the Cauchy-Schwarz inequality and Assumption 4.3, we have

$$
\begin{aligned}
\mathbb{E}\left\|\sum_{b=1}^{B}(\frac{1}{B}-\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}}{\sum_{n\in\mathcal{M}}q_n})\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2&\leq B\mathbb{E}\left[\sum_{b=1}^{B}(\frac{1}{B}-\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}}{\sum_{n\in\mathcal{M}}q_n})^2\left\|\frac{1}{B}\sum_{b=1}^{B}\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2\right]\\
&=B\|\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\|^2\mathbb{E}[QCID(\mathcal{M})]+\gamma^2\mathbb{E}[QCID(\mathcal{M})]
\end{aligned}
\tag{10}
$$

For $T_4$, we have

$$
\begin{aligned}
&\mathbb{E}\left\|\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}[\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)-\boldsymbol{h}_{(n,b)}^{(k)}]}{\sum_{n\in\mathcal{M}}q_n}\right\|^2\\
&=\mathbb{E}\left\|\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}[\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)-\nabla F_{(n,b)}\left(\boldsymbol{w}^{(k,0)}\right)+\nabla F_{(n,b)}\left(\boldsymbol{w}^{(k,0)}\right)-\boldsymbol{h}_{(n,b)}^{(k)}]}{\sum_{n\in\mathcal{M}}q_n}\right\|^2\\
&\leq 2\mathbb{E}\left\|\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}[\nabla\widetilde{F}_b\left(\boldsymbol{w}^{(k,0)}\right)-\nabla F_{(n,b)}\left(\boldsymbol{w}^{(k,0)}\right)]}{\sum_{n\in\mathcal{M}}q_n}\right\|^2 \qquad (11)\\
&+2\mathbb{E}\left\|\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}[\nabla F_{(n,b)}\left(\boldsymbol{w}^{(k,0)}\right)-\boldsymbol{h}_{(n,b)}^{(k)}]}{\sum_{n\in\mathcal{M}}q_n}\right\|^2\\
&\leq 2\kappa^2\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}}{\sum_{n\in\mathcal{M}}q_n}+2\left\|\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}[\nabla F_{(n,b)}\left(\boldsymbol{w}^{(k,0)}\right)-\boldsymbol{h}_{(n,b)}^{(k)}]}{\sum_{n\in\mathcal{M}}q_n}\right\|^2\\
&\leq 2\kappa^2+2\left\|\sum_{b=1}^{B}\frac{\sum_{n\in\mathcal{M}}q_n\alpha_{(n,b)}[\nabla F_{(n,b)}\left(\boldsymbol{w}^{(k,0)}\right)-\boldsymbol{h}_{(n,b)}^{(k)}]}{\sum_{n\in\mathcal{M}}q_n}\right\|^2, \qquad (12)
\end{aligned}
$$

where where $\kappa = max_{\{n,b\}}\kappa_{\{n,b\}}$. According to the results from the proof in C.5 in (Wang et al., 2020a), we have

$$\begin{aligned}
&\mathbb{E}\left\|\nabla F_{(n,b)}\left(\boldsymbol{w}^{(k,0)}\right)-\boldsymbol{h}_{(n,b)}^{(k)}\right\|^2 \leq \frac{L_{n,b}^2}{\tau}\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\boldsymbol{w}^{(k,0)}-\boldsymbol{w}_n^{(k,t)}\right\|^2\right] \\
&\leq \frac{L^2}{\tau}\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\boldsymbol{w}^{(k,0)}-\boldsymbol{w}_n^{(k,t)}\right\|^2\right] \\
&\leq \frac{2\eta^2L^2\sigma^2}{1-D}(\tau-1)+\frac{D}{1-D}\mathbb{E}\left[\left\|\nabla F_i\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2\right] \\
&\leq \frac{2\eta^2L^2\sigma^2}{1-D}(\tau-1)+\frac{2D}{1-D}\left\|\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2+\frac{2D}{1-D}\mathbb{E}\left[\left\|\nabla F_i\left(\boldsymbol{w}^{(k,0)}\right)-\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2\right] \\
&\leq \frac{2\eta^2L^2\sigma^2}{1-D}(\tau-1)+\frac{2D}{1-D}\left\|\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2+\frac{2D}{1-D}\mathbb{E}\left[\left\|\nabla F_i\left(\boldsymbol{w}^{(k,0)}\right)-\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2\right] \\
&\leq \frac{2\eta^2L^2\sigma^2}{1-D}(\tau-1)+\frac{2D}{1-D}\left\|\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2+\frac{1}{B}\frac{2D}{1-D}\kappa^2
\end{aligned} \tag{13}$$

where $L = max_{\{n,b\}}L_{n,b}$ and $D = 4\eta^2L^2\tau(\tau-1)$.

Combining the results in (8), (9), (10), (12) and (13), it is easy to derive that

$$\begin{aligned}
&\mathbb{E}\left[\widetilde{F}\left(\boldsymbol{w}^{(k+1,0)}\right)\right]-\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right) \\
\leq &-(\frac{1}{2}-B\delta\mathbb{E}[QCID]-\frac{4D}{1-D})\eta\left\|\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2+4\eta\kappa^2+\frac{4\eta^3L^2\sigma^2}{1-D}(\tau-1) \\
&+\frac{1}{B}\frac{4D}{1-D}\eta\kappa^2+\eta^2L_{\widetilde{F}}\sigma^2+\gamma^2\eta\mathbb{E}[QCID]
\end{aligned} \tag{14}$$

Now we have

$$\begin{aligned}
&\frac{\mathbb{E}\left[\widetilde{F}\left(\boldsymbol{w}^{(k+1,0)}\right)\right]-\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)}{\eta} \\
\leq &-(\frac{1}{2}-B\delta\mathbb{E}[QCID]-\frac{4D}{1-D})\left\|\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2+4\kappa^2 \\
&+\frac{4\eta^2L^2\sigma^2}{1-D}(\tau-1)+\frac{1}{B}\frac{4D}{1-D}\kappa^2+\gamma^2\mathbb{E}[QCID]+\eta L_{\widetilde{F}}\sigma^2
\end{aligned} \tag{15}$$

Taking the total expectation and averaging over all rounds, one can obtain

$$\begin{aligned}
&\frac{\mathbb{E}\left[\widetilde{F}\left(\boldsymbol{w}^{(K,0)}\right)\right]-\widetilde{F}\left(\boldsymbol{x}^{(0,0)}\right)}{\eta K} \leq -(\frac{1}{2}-B\delta\mathbb{E}[QCID]-\frac{4D}{1-D})\frac{1}{K}\sum_{t=1}^{K-1}\mathbb{E}\left\|\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2 \\
&\gamma^2\mathbb{E}[QCID]+(4+\frac{1}{B}\frac{4D}{1-D})\kappa^2+\frac{4\eta^2L^2\sigma^2}{1-D}(\tau-1)+\eta L\sigma^2
\end{aligned} \tag{16}$$

Finally, we have

$$\begin{aligned}
\min_{k\leq K}\left\|\nabla F\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2 \leq \frac{1}{K}\sum_{k=1}^{K-1}\mathbb{E}\left\|\nabla\widetilde{F}\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2 \leq &\frac{1}{(\frac{1}{2}-B\delta\mathbb{E}[QCID(\mathcal{M})]-\frac{4D}{1-D})}[\frac{\widetilde{F}\left(\boldsymbol{w}^{(0,0)}\right)-\widetilde{F}_{min}}{\eta K} \\
&+(4+\frac{1}{B}\frac{4D}{1-D})\kappa^2+\frac{4\eta^2L^2\sigma^2}{1-D}(\tau-1)+\eta L_{\widetilde{F}}\sigma^2+\gamma^2\mathbb{E}[QCID]]
\end{aligned} \tag{17}$$

If setting $\eta = \frac{s}{10L\sqrt{\tau(\tau-1)K}}$ with $s < 1$, we have

$$\min_{k\leq K}\left\|\nabla F\left(\boldsymbol{w}^{(k,0)}\right)\right\|^2 \leq \frac{1}{\frac{1}{3} - B\delta\mathbb{E}[QCID(\mathcal{M})]}[\frac{\widetilde{F}\left(\boldsymbol{w}^{(0,0)}\right) - \widetilde{F}_{min}}{s\sqrt{K}/(10L\sqrt{\tau(\tau-1)})} + 5\kappa^2 + \frac{\sigma^2 s^2}{25\tau K} + \frac{sL_{\widetilde{F}}\sigma^2}{10L\sqrt{\tau(\tau-1)K}} + \gamma^2\mathbb{E}[QCID]] \tag{18}$$

Since $\widetilde{F}$ is larger than 0, $F_{min} > 0$. Now we let $\boldsymbol{w}^{(k)}$ denote the global model parameter at the $k$-th communication round and $\boldsymbol{w}^{(0)}$ denote the initial parameter. After changing the notations, we can finish our proof by the following:

$$\min_{k\leq K}\left\|\nabla \widetilde{F}\left(\boldsymbol{w}^{(k)}\right)\right\|^2 \leq \frac{1}{\frac{1}{3} - B\delta\mathbb{E}[QCID(\mathcal{M})]}[\frac{\sigma^2 s^2}{25\tau K} + \frac{sL_{\widetilde{F}}\sigma^2}{10L\sqrt{\tau(\tau-1)K}} + 5\kappa^2 + \frac{10L\sqrt{\tau(\tau-1)}\widetilde{F}\left(\boldsymbol{w}^{(0)}\right)}{s\sqrt{K}} + \gamma^2\mathbb{E}[QCID]],$$

□

## C. Supplemental Experiment Settings and Results

### C.1. The Experimental Settings in Section 2.1

We adopt an MLP model with one hidden layer of 64 units and FedAvg (McMahan et al., 2017b) as the FL optimizer. In Figure 1a, we allocate the MNIST data to $N = 100$ clients with each client only accessing to the same amount of data from one class. In Figure 1b, each client is associated with the same amount of data from two classes. In Figure 1c and 1d, we first allocate the whole MNIST dataset to $N = 200$ clients and pick 100 to construct a class-imbalanced global dataset. The global dataset with the 100 clients has the same amount of $n_1$ data samples for five classes and has the same amount of $n_2$ data samples for the other five classes. The ration r between $n_1$ and $n_2$ is set to $3:1$.

In each training round (communication round), all of the clients conduct 5 local training epochs. The batch size is 50 for each client. The local optimizer is SGD with a weight decay of 0.0005. The learning rate is 0.01 initially and the decay factor is 0.9992. We terminate the FL training after 200 training rounds (communication rounds) and then evaluate the model's performance on the test dataset of MNIST.

### C.2. Additional Experimental Settings in Section 5

The model we adopt has two convolutional layers with the number of kernels being 6 and 16, respectively. And all convolution kernels are of size $5 \times 5$. The outputs of convolutional layers are fed into two hidden layers with 120 and 84 units.

In our implementation of Power-of-choice selection strategy (pow-d)(Cho et al., 2020), we first sample a candidate set $\mathcal{A}$ of 20 clients without replacement such that client $n$ is chosen with probability proportional to the size of their local dataset $q_n$. Then the server sends the current global model to the clients in set $\mathcal{A}$, and these clients compute and send back to the server their local loss. To derive $\mathcal{M}$, we select M clients who have the highest loss from $\mathcal{A}$.

In our implementation of the method in (Yang et al., 2020) (Fed-cucb), the exploration factor to balance the trade-off between exploitation and exploration is set as 0.2 and the forgetting factor as 0.99, which is the same as the settings in (Yang et al., 2020).

With the help of FHE, we can derive the matrix of inner products $S$ accurately. Hence, in the simulation of our method, Fed-CBS, we ignore the process of deriving $S$ and focus on our sampling strategy.

### C.3. Additional Details for the Experimental Settings in Case 1 and Case 2

**Case 1** In this setting, we have 120 clients in total, and each client has only one class of data.

When $n_1 : n_2 = 3 : 1$, there are 18 clients having the data from the 1st class, 18 clients having the data from the 2nd class, 18 clients having the data from the 3rd class, 18 clients having the data from the 4th class, and 18 clients having the data

from the 5th class. There are 6 clients with data from the 6th class, 6 clients with data from the 7th class, 6 clients with data from the 8th class, 6 clients with data from the 9th class, and 6 clients the data from the 10th class.

When $n_1 : n_2 = 5 : 1$, there are 20 clients having the data from the 1st class, 20 clients having the data from the 2nd class, 20 clients having the data from the 3rd class, 20 clients having the data from the 4th class and 20 clients having the data from the 5th class. There are 4 clients having the data from the 6th class, 4 clients having data from the 7th class, 4 clients having data from the 8th class, 4 clients having data from the 9th class, and 4 clients having the data from the 10th class.

Then we uniformly set 30% (36 clients) of them available. Since there are more clients which contain the data from the first 5 classes among the above 120 clients. The global dataset of these 36 clients is often class-imbalanced.

**Case 2** In this setting, we have 200 clients in total and each client has only one class of data. For all the $i \in \{1, 2, ..., 10\}$, there are 20 clients having the data from the $i-$th class.

When $n_1 : n_2 = 3 : 1$, we randomly pick 9 clients from the 20 clients which have the data from the 1st class and set them available. We randomly pick 9 clients from the 20 clients which have the data from the 2nd class and set them available. Similarly, for the $k$-th class ($2 < k \leq 5$), we randomly pick 9 clients from the 20 clients which have the data from the $k$-th class and set them available. On the contrary, we randomly pick 3 clients from the 20 clients which have the data from the 6th class and set them available. We randomly pick 3 clients from the 20 clients which have the data from the 7th class and set them available. Similarly, for $7 < k \leq 10$, we randomly pick 3 clients from the 20 clients which have the data from the $k$-th class and set them available. There are 60 clients in total.

When $n_1 : n_2 = 5 : 1$, we randomly pick 10 clients from the 20 clients which have the data from the 1st class and set them available. We randomly pick 10 clients from the 20 clients which have the data from the 2nd class and set them available. For the $k$-tth class ($2 < k \leq 5$), we randomly pick 10 clients from the 20 clients which have the data from the $k$-th class and set them available. On the contrary, we randomly pick 2 clients from the 20 clients which have the data from the 6th class and set them available. We randomly pick 2 clients from the 20 clients which have the data from the 7th class and set them available. And for the other $k$-th class ($7 < k \leq 10$), we randomly pick 2 clients from the 20 clients which have the data from the $k$-th class and set them available. There are 60 clients in total.

Since there are more clients that contain the data from the first 5 classes among the above 60 clients, the global dataset of these 60 clients is always class-imbalanced.

The difference between the settings of Case 1 and Case 2 is that we uniformly set 30% clients available in Case 1 but non-uniformly set 30% clients available in Case 2. Nevertheless, the global datasets of the available clients are both class-imbalanced in both cases.

### C.4. The Averaged QCID Values for Case 1 and Case 2 in Section 5.2

| $\mathbb{E}[QCID](10^{-2})$ | | all | rand | pow-d | Fed-cucb | Fed-CBS |
|---|---|---|---|---|---|---|
| Case 1 | 3:1 | 2.90±0.02 | 9.33±0.17 | 13.70±0.39 | 1.39±0.37 | **0.57±0.04** |
| | 5:1 | 6.17±0.04 | 12.36±0.20 | 16.63±0.74 | 3.43±0.76 | **2.41±0.07** |
| Case 2 | 3:1 | 2.50±0.00 | 9.91±0.16 | 13.68±0.72 | 1.89±1.72 | **0.001±0.001** |
| | 5:1 | 4.44±0.00 | 11.70±0.20 | 15.68±0.96 | 2.63±2.40 | **0.002±0.001** |

*Table 3.* The averaged $QCID$ values for four baselines and our method. Our method, Fed-CBS, has successfully reduced the class-imbalance. Since the global dataset of all the 60 available clients is always class-imbalanced and the ratio is always fixed in case 2, the $QCID$ value is fixed and the derivation of it is always zero.

### C.5. Experiment Results of Fashion-MNIST Dataset

**Experiment Setup** We adopt an MLP model with one hidden layer of 64 units and and FedNova (Wang et al., 2020a) as the FL optimizer . Similar to the setup in the experiment of CIFAR-10, the batch size is 50 for each client. In each communication round, all of them conduct the same number of local updates, which allows the client with the largest local dataset to conduct 5 local training epochs. In our method, we set the $\beta_m = m$, $\gamma = 10$ and $L_b = 10^{-20}$. The local optimizer is SGD with a weight decay of 0.0005. The learning rate is 0.01 initially and the decay factor is 0.9992. We terminate the FL training after 3000 communication rounds and then evaluate the model's performance on the test dataset of Fashion-MNIST.

| | | all | rand | pow-d | Fed-cucb | Fed-CBS |
|---|---|---|---|---|---|---|
| Communication Rounds | $\alpha$=0.1 | 115±17 | 185±27 | 135±22 | 124±37 | **92±6** |
| | $\alpha$=0.2 | 173±45 | 284±54 | 218±55 | 216±24 | **166±36** |
| | $\alpha$=0.5 | 258±44 | 331±55 | 281±54 | 284±51 | **218±36** |
| $\mathbb{E}[QCID](10^{-2})$ | $\alpha$=0.1 | 1.40±0.11 | 8.20±0.19 | 11.72±0.33 | 4.24±0.59 | **0.15±0.02** |
| | $\alpha$=0.2 | 1.39±0.22 | 7.67±0.26 | 10.31±0.24 | 4.43±0.38 | **0.21±0.01** |
| | $\alpha$=0.5 | 0.94±0.07 | 5.93±0.26 | 7.68±0.28 | 4.34±0.85 | **0.22±0.01** |

*Table 4.* The communication rounds required for targeted test accuracy and the averaged QCID values on Fashion-MNIST dataset. The targeted test accuracy is 78% for $\alpha = 0.1$, 80% for $\alpha = 0.2$ and 82% for $\alpha = 0.5$. The results are the mean and the standard deviation over 4 different random seeds.

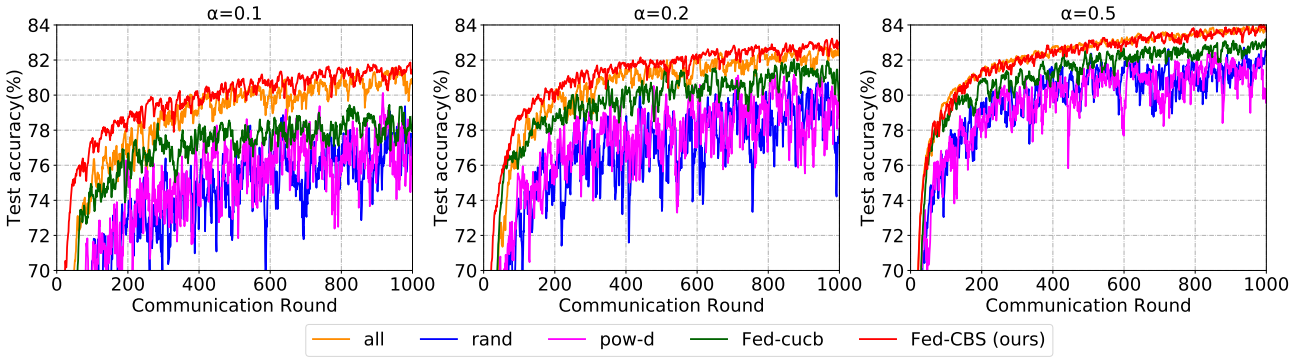



*Figure 8.* Test accuracy on Fashion-MNIST dataset under three heterogeneous settings.

### C.5.1. Results for Class-Balanced Global Dataset

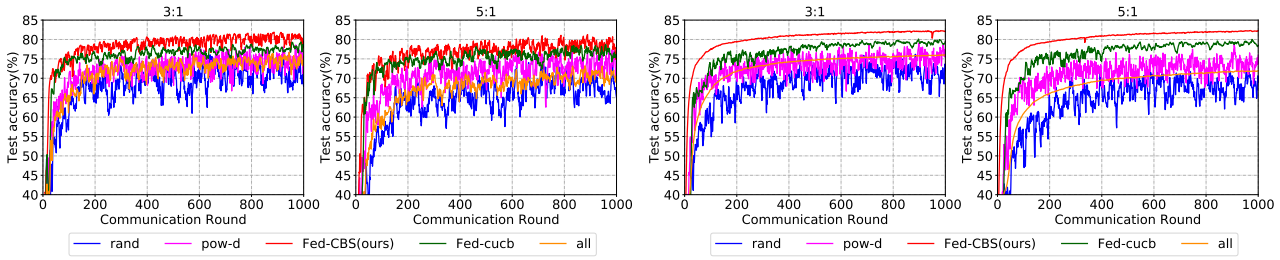
Similar to the experiment settings, in this experiment, we set 200 clients in total with a class-balanced global dataset. The non-IID data partition among clients is based on the settings of Dirichlet distribution parameterized by the concentration parameter $\alpha$ in (Hsu et al., 2019). In each communication round, we uniformly and randomly set 30% of them (i.e., 60 clients) available and select 10 clients from those 60 available ones to participate in the training.

As shown in Table 8, our method successfully reduces the class-imbalance, since it achieves the lowest $QCID$ value compared with other client selection strategies. Our method outperforms the other three baseline methods and achieves comparable performance in the ideal setting where all the available clients are engaged in the training. As shown in Table 3 and Figure 8, our method can achieve faster and more stable convergence. It is worth noting that due to the inaccurate estimation of distribution and the weakness of the greedy method discussed in Section 2.2, the performance of Fed-cucb is much worse than ours.

### C.5.2. Results for Class-Imbalanced Global Dataset: Case 1

Similar to the settings for Cifar-10, there are 120 clients in total and each client only has one class of data with the same quantity. The global dataset of these 120 clients is always class-imbalanced. To measure the degree of class imbalance, we let the global dataset have the same amount $n_1$ of data samples for five classes and have the same amount $n_2$ of data samples for the other five classes. The ratio $r$ between $n_1$ and $n_2$ is set to $3:1$ and $5:1$ respectively in the experiments. In each communication round, we randomly set 30% of them (*i.e.*, 36 clients) available and select 10 clients to participate in the training.

As shown in the Table 5 and Figure 9a, our method can achieve faster and more stable convergence, and even better performance than the ideal setting where all the available clients are engaged. The performance of Fed-cucb (Yang et al., 2020) is better than the results on class-balanced global dataset, which is partly due to the simplicity of each client's local dataset composition in our experiments as discussed in the experiments of Cifar-10.




(a) Case 1 (b) Case 2

*Figure 9.* Test accuracy on Fashion-MINST with class-imbalanced global dataset in Case 1 and Case 2.

| | | all | rand | pow-d | Fed-cucb | Fed-CBS |
|---|---|---|---|---|---|---|
| Case 1 | 3:1 | 78.42±0.79 | 78.46±0.90 | 81.08±0.21 | 80.83±0.91 | **81.75±0.34** |
| | 5:1 | 72.42±2.22 | 75.49±2.56 | 80.15±0.41 | 80.50±0.95 | **81.42±0.50** |
| Case 2 | 3:1 | 74.64±1.87 | 78.80±0.55 | 81.13±0.41 | 79.94±0.31 | **81.95±0.57** |
| | 5:1 | 67.16±4.13 | 74.17±2.01 | 80.05±0.39 | 80.00±0.58 | **81.92±0.57** |

*Table 5.* Best test accuracy for our method and other four baselines on Fashion-MNIST dataset.

#### C.5.3. Results for Class-Imbalanced Global Dataset: Case 2

Similar to the settings of Cifar-10, we assume that there are 200 clients in total. In each communication round, 30% of them (*i.e.*, 60 clients) are set available in each training round. The global dataset of those 60 available clients is always class-imbalanced. To measure the degree of class imbalance, we make the global dataset have the same amount $n_1$ of data for the five classes and have the same amount $n_2$ of data for the other five classes. The ratio $r$ between $n_1$ and $n_2$ is set to $3:1$ and $5:1$. We select 10 clients from these 60 clients to participate in the training.

As shown in the Table 5 and Figure 9b, our method can achieve higher test accuracy and more stable convergence, which outperforms the ideal setting where all the available clients are engaged. Since the global dataset of the available 60 clients in each communication round is always class-imbalanced, the performance of engaging all of them is not good.

## D. Ablation Studies and Discussion

### D.1. Accurate Estimation vs Inaccurate Estimation for Fed-cucb

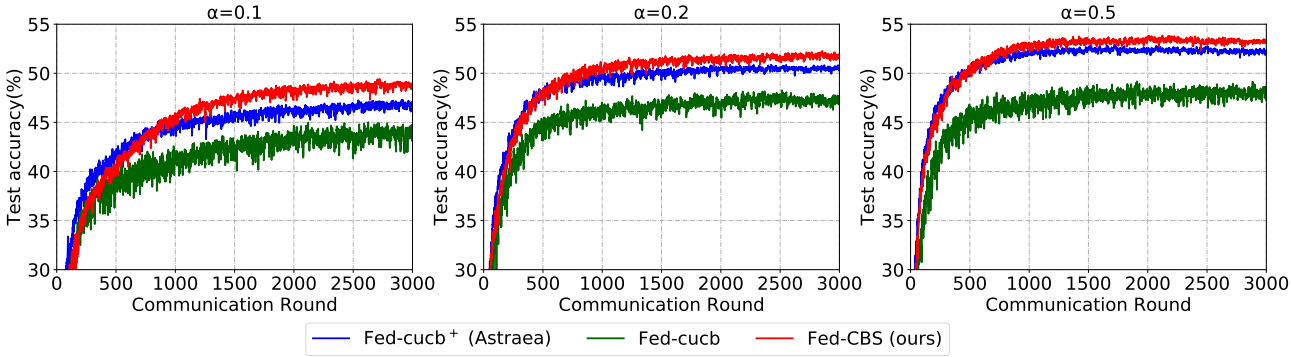



*Figure 10.* Test accuracy on Cifar-10 for Fed-cucb, Fed-cucb$^+$ and Fed-CBS.

As discussed in Sections 2.2 and 5.1, the estimation of the label distribution in Fed-cucb (Yang et al., 2020) is not accurate, which leads to performance degradation. Hence there comes a natural question, would the performance of Fed-cucb get improved if it got an exact estimation of the local label distribution? In our simulation, we manually let the Fed-cucb know the exact value of each client's local label distribution and name it as Fed-cucb$^+$. Actually, Fed-cucb$^+$ is the core part

| | | Fed-cucb$^+$ (Astraea) | Fed-cucb | Fed-CBS |
|---|---|---|---|---|
| Best Accuracy (%) | $\alpha$=0.1 | 49.10±0.70 | 46.84±0.73 | **50.36±0.58** |
| | $\alpha$=0.2 | 50.61±0.77 | 48.80±1.05 | **51.95±0.57** |
| | $\alpha$=0.5 | 52.71±0.27 | 50.98±0.56 | **54.21±0.34** |
| $\mathbb{E}(QCID)$ ($10^{-2}$) | $\alpha$=0.1 | 0.83±0.18 | 7.09±2.27 | **0.62±0.20** |
| | $\alpha$=0.2 | 0.68±0.05 | 5.93±1.01 | **0.51±0.12** |
| | $\alpha$=0.5 | 0.43±0.04 | 6.47±0.77 | **0.36±0.04** |

*Table 6.* Best accuracy and the averaged $QCID$ values.

of Astraea (Duan et al., 2019) without data augmentation. Hence, comparing our method with Fed-cucb$^+$ can show the superiority of our sampling strategy over the greedy method in Fed-cucb (Yang et al., 2020) and Astraea (Duan et al., 2019).

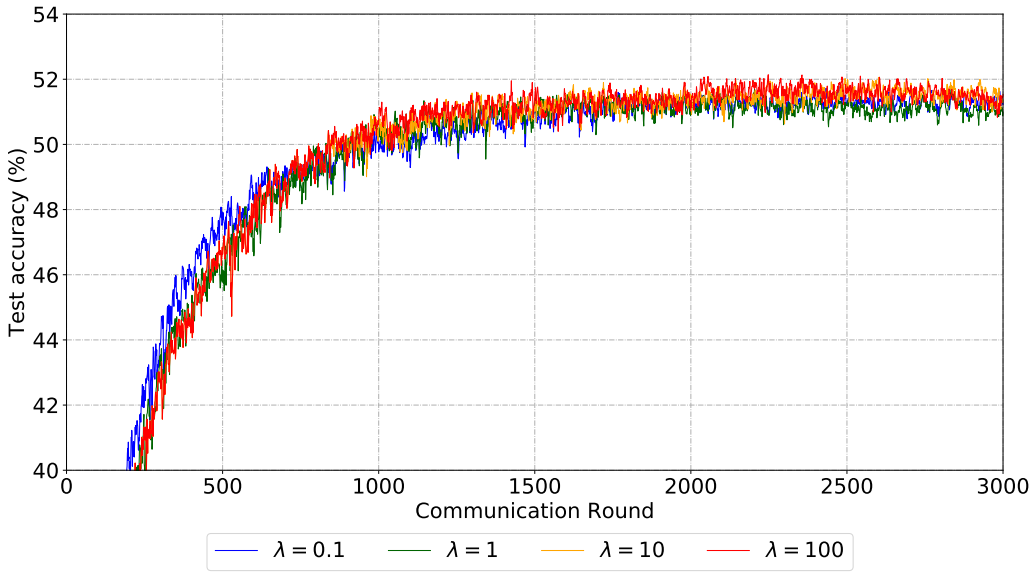



*Figure 11.* Test accuracy with different exploration factor $\lambda$.

### D.2. The Effect of Exploration Factor $\lambda$

As shown in the Figure 10 and Table 6, Fed-cucb$^+$ does improve the performance of Fed-cucb, which verifies the importance of accurate estimation. However, our Fed-CBS still outperforms Fed-cucb$^+$. Although, it seems that the accuracy of Fed-cucb$^+$ increases a little faster than Fed-CBS at the beginning of the training, our method will achieve higher accuracy as the training proceeds further. As discussed in the Remark 3.3 in Section 3.3 and the Figure 3 of Section 5.1, this is due to the pitfall of greedy method, where one will miss the optimal solution. This has been verified by the averaged $QCID$ value in Table 6, which shows that Fed-CBS can achieve lower $\mathbb{E}(QCID)$ than Fed-cucb$^+$ (Astraea).

Another potential weakness of greedy method is the diversity of client composition. Following their selection process, once the first choice of client has been made, the following choices are fixed successively. Hence there are only limited kinds of client composition. It is interesting to investigate the relationship between the training performance and the diversity of client composition and we leave it as future work.

In our sampling strategy, when we sample the first client, we introduce the exploration factor $\lambda$ to balance the tradeoff between exploitation and exploration. When the $\lambda$ is small, our method will tend to exploit the class-balanced clients since their $QCID$ values are smaller. For fairness, we hope every client can get the chance to be selected. Hence, we can increase the $\lambda$ and then our method will tend to explore the clients which have seldom been selected before. However, it might cost many communication rounds for exploration and lead to slower convergence.

We conduct some experiments to verify the effect of exploration factor $\lambda$. The settings are the same as the ones in Section

5.1 when $\alpha = 0.2$. As shown in the Figure 11, as the $\lambda$ becomes larger, the increase of accuracy will become a little slower at the start of the training. This because the it might cost more communication rounds for exploration. As the training proceeds, the accuracy with larger $\lambda$ becomes a little higher than the ones with smaller $\lambda$. Overall, the improvement on the convergence speed and best accuracy is very slight, which means the performance of FL training is not very sensitive to the values of exploration factor $\lambda$. Generally, if we want to slightly fasten the convergence, we can decrease the value of $\lambda$. If we want to improve the best accuracy a little, we can increase the value of $\lambda$.

### D.3. The Performance with Different Amounts of Selected Clients

In this section, we want to investigate how the amount of selected clients will affect the FL training performance. Generally, we think as the amount of selected clients increases, the FL training process can achieve better performance. However, once that amount reaches some threshold $\epsilon$, the improvement will become slighter. This is because we find that select only a subset of all the available can achieve comparable results with engaging all the available clients into the training. As for how to decide the threshold $\epsilon$, we provide the following two principles based on $QCID$ and our experience.

- First, if we work on a classification task with $B$ classes, we can select at least B clients. This is because in some special cases, each client will only have one class of data in their local dataset, such as the settings in Section 5.2. Hence, if less than B clients are selected, the grouped dataset of the selected clients will miss some classes of data.
- Second, to avoid missing some classes of data, we increase the threshold $\epsilon$ such that the averaged $QCID$ value could be smaller than $\frac{1}{B^2}$. This is because if the grouped dataset misses at least one class of data, the $QCID$ will be larger than $\frac{1}{B^2}$.

We conducted some experiments to verify our prediction on the effect of the amount of selected clients. The settings are the same as the ones in Section 5.1 when $\alpha = 0.2$. As shown in the left figure of Figure 12, as the amount of selected clients increases, the FL training process can achieve better performance. However, when the amount M is larger than 10, the improvement is slighter. In the right figure of Figure 12 , we can find that the averaged $QCID$ value of selecting 5 clients is larger than $(\frac{1}{10})^2 = 0.01$ and its performance is obviously worse than the others. These results verify the effectiveness of our principles on how to set the threshold $\epsilon$. It is worth noting that due to the limitation of communication capacities, we cannot select as many clients as possible. Hence, how to identify the appropriate threshold $\epsilon$ is critical to the FL training.

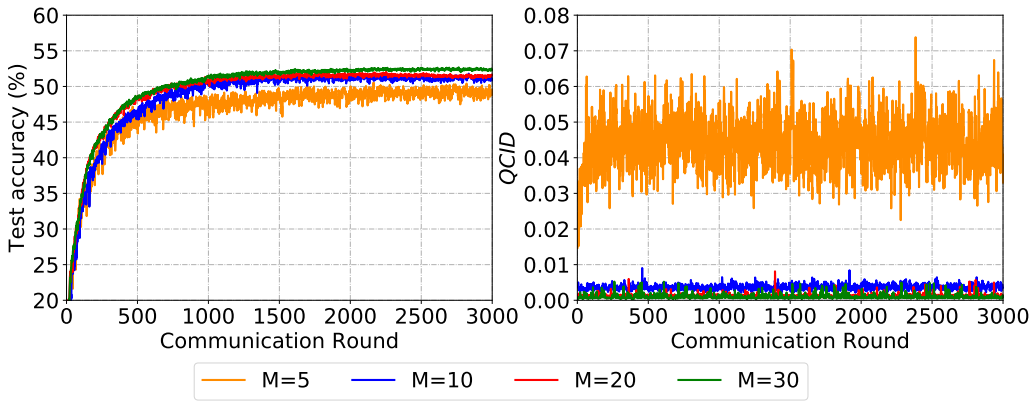



*Figure 12.* Left: The performance with different amounts of selected clients. Right: The $QCID$ with different amounts of selected clients.

### D.4. Additional Experimental Results on FEMNIST Dataset

We also conduct some experiments on the FEMINST Dataset to simulate more realistic settings where there are thousands of clients. Since in practice, it is impossible to engage all the clients during training, we compare our method by randomly selecting more clients. . There are 3500 ($> 1000$) clients in total and we randomly set $10\%(< 30\%)$ of them available in each round. Then our method tries to select 30 clients from them. That is less than $1\%$ of all the 3550 clients and also less than the number of classes (64). Besides, we also run three baselines, randomly selecting 30 clients, randomly selecting

120($>$ 100) clients, and selecting 30 clients with fed-cucb. We present the results in Table 7 and Figure 13. Our performance is still the best. Due to the global imbalance, the rand-120 is even worse than the rand-30.

| | rand-30 | rand-120 | Fed-cucb | Fed-CBS |
|---|---|---|---|---|
| Communication Rounds | 1106 $\pm$ 24 | 1394 $\pm$ 11 | 1124$\pm$ 31 | **980$\pm$17** |

*Table 7.* The communication rounds required for targeted test accuracy (75%). The results are the mean and the standard deviation over 3 different random seeds.

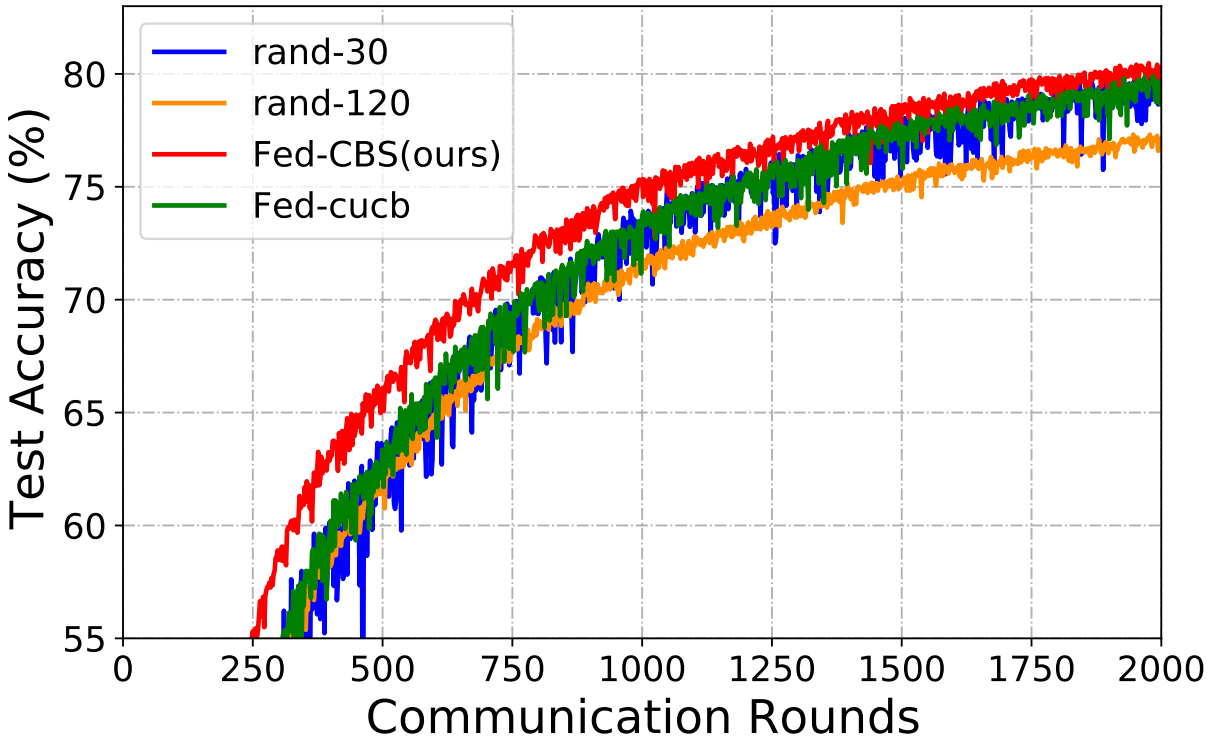



*Figure 13.* Test accuracy for FEMNIST

## E. Comparison between Cluster-based Client Sampling Algorithms and Fed-CBS

We present the following comparison between cluster-based client sampling algorithms and our own method to demonstrate our superiority.

Firstly, the unbiased sampling property of the clustering sampling method (Fraboni et al., 2021) may not lead to optimal performance when dealing with class-imbalanced global training datasets. In Section 3.1 of (Fraboni et al., 2021), the authors mention that they "require clustered sampling to be unbiased," which implies that the expected value of client aggregation should be equivalent to the aggregation of all clients. However, our findings, as depicted in Figures 1, indicate that aggregating all clients does not always lead to satisfactory performance, especially when the downstream test task is class-balanced. It should be noted that ensuring class-balance in the downstream test task is crucial for maintaining fairness and privacy. This is because the imbalanced performance of the model across different classes can potentially reveal sensitive information about the global training dataset.

Secondly, our method guides the clustering sampling methods. Although clustering sampling can address many root causes of heterogeneity in the input space distributions at clients, however, since "unbiased sampling" will cause the mismatch between the input space distributions at clients and the downstream task's space distribution, we still need to identify key causes to make the clustering sampling "biased" to align the input space and downstream space. This is still very challenging because while clustering sampling methods can include many root causes of heterogeneity in the input space distributions at

clients, we still need to be careful since most are hard to measure and contain lots of private information. Our analysis of "class imbalance" provides a valuable measure in this regard, and we also offer an efficient means of utilizing this measure in a privacy-preserving way. Therefore, our work can contribute to advancing clustering sampling methods in the future