FedMoS: Taming Client Drift in Federated Learning with Double Momentum and Adaptive Selection

Xiong Wang*, Yuxin Chen*, Yuqing Li†, Xiaofei Liao*, Hai Jin*, and Bo Li‡
* National Engineering Research Center for Big Data Technology and System,
Services Computing Technology and System Lab, Cluster and Grid Computing Lab,
School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
† School of Cyber Science and Engineering, Wuhan University, Wuhan, China
‡ Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong
E-mail: *{xiongwang, yuxinchen, xfliao, hjin}@hust.edu.cn, †li.yuqing@whu.edu.cn, ‡bli@cse.ust.hk

IEEE INFOCOM 2023 - IEEE Conference on Computer Communications | 979-8-3503-3414-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/INFOCOM53939.2023.10228957

***Abstract*—*Federated learning* (FL) enables massive clients to collaboratively train a global model by aggregating their local updates without disclosing raw data. Communication has become one of the main bottlenecks that prolongs the training process, especially under large model variances due to *skewed* data distributions. Existing efforts mainly focus on either single momentum-based gradient descent, or random client selection for potential variance reduction, yet both often lead to poor model accuracy and system efficiency. In this paper, we propose FedMoS, a communication-efficient FL framework with *coupled* double momentum-based update and adaptive client selection, to jointly mitigate the intrinsic variance. Specifically, FedMoS maintains *customized momentum buffers* on both server and client sides, which track global and local update directions to alleviate the model discrepancy. Taking momentum results as input, we design an *adaptive selection scheme* to provide a proper client representation during FL aggregation. By optimally calibrating clients' selection probabilities, we can effectively reduce the sampling variance, while ensuring unbiased aggregation. Through a rigid analysis, we show that FedMoS can attain the *theoretically optimal* $\mathcal{O}(T^{-2/3})$ convergence rate. Extensive experiments using real-world datasets further validate the superiority of FedMoS, with *58%-87% communication reduction* for achieving the same target performance compared to state-of-the-art techniques.**

I. Introduction

Recent years have witnessed an enormous success achieved by machine learning technology. However, a long-standing concern remains with potential leakage of user privacy, where the data samples utilized to train machine learning models may contain users' sensitive and confidential information. Due to such privacy concern, *federated learning* (FL) has been widely recognized as a promising paradigm which facilitates massive clients (e.g., edge devices) to collaboratively build a global model without centralizing training data [1]. In a typical FL system, there is a central parameter server orchestrating geo-distributed clients to aggregate their local models in multiple rounds of synchronization. Thus far, a wide spectrum of FL applications has been deployed, such as keyword prediction [2], image classification [3], and human activity recognition [4].

Communication often becomes the primary bottleneck for FL due to frequent model synchronization. Even worse, connections between many edge clients and central server can be intermittent and unstable, while the constrained server capacity also imposes restrictions on the number of clients a server can simultaneously accommodate. Both lead to a growing interest in *reducing communication* during federated training. In particular, FedAvg, a de facto communication-efficient FL algorithm, was proposed by requesting clients to perform multiple steps of local *stochastic gradient descent* (SGD) before sending their updates, i.e., to decrease synchronizing frequency at the cost of increased computation on clients [5]. However, unlike datacenter based distributed learning, FL prohibits collecting and shuffling data beforehand so that training sample distributions across isolated clients are highly skewed and non-i.i.d. [6]. This will cause local model updates to gradually diverge, a.k.a. client drift, thereby leading to biased global model with large variance and prolonging the training process. How to enhance the *communication efficiency under client drift* is of paramount importance to the FL community.

A surge of interest has been attracted to nested variance reduction for accelerating training convergence and thus enhancing the communication efficiency. One popular workaround is to employ the *momentum-based SGD optimization* to smooth out the noise of stochastic gradients [7]. In general, a client or server can maintain a momentum buffer to track the stochastic updates, where the influence of previous update directions is preserved when renewing model parameters [8], [9]. This way, a momentum SGD enjoys a faster convergence with reduced inherent variance, which greatly reduces the communication. Despite potential benefits of momentum design, *cherry-picking a subset of representative clients* participating in FL is equally, if not more important, to improve the training convergence and model accuracy [10]. Considering the existence of selection bias, it is desirable to choose the best client subset for sampling variance reduction. Although many efforts have been devoted to momentum implementation and client sampling, they were largely explored *separately*. In broad strokes, momentum

The research was supported in part by National Key Research and Development Program of China under grant 2022ZD0115301, by NSFC under grant 62202185, and by a RGC RIF grant under the contract R6021-20, and RGC GRF grants under the contracts 16209120 and 16200221. (Corresponding author: Yuqing Li.)

SGD often involves local updates on each selected client, meanwhile the optimal sampling decision needs to be made based on momentum results. Therefore, it is imperative to *jointly address these two issues together* to achieve favorable FL convergence and accuracy. This, however, is challenging due to the following reasons.

First and foremost, the inter-dependence between momentum SGD and client selection requires an *integrated design* in the FL framework, rather than a separate characterization in previous works [7], [11]. Nevertheless, existing selection schemes are mostly implemented atop of *plain gradient-based algorithms* (FedAvg [5], FedProx [6], etc.). A joint optimization of momentum and selection remains largely *unexplored*. Worse yet, bias may coexist with the model update and client sampling [12], which will hinder the learning performance. As a result, it is necessary to propose a *co-design* of momentum-based update and client selection for ensuring *unbiasedness*, meanwhile collectively reducing the intrinsic model variance. Second, FL includes stochastic updates on both sides of the central server and distributed clients, which makes existing momentum solutions, mainly proposed for single-side gradient direction tracking, *insufficient* to accomplish a favorable training speedup. Besides, double momentum design requires a careful coordination between global and local updates to strike the *orderly optimal* convergence result, yet may still behave poorly when facing extreme client drift. This demands more practical countermeasures in the momentum implementation, while also achieving a rigorous theoretical guarantee. Third, to derive the optimal selection scheme usually needs many *extra* operations (e.g., communications and computations) to acquire the client information for a wiser client sampling [13], [14], which however can be unduly expensive in practice. How to optimize the selection cost, even *avoiding* those additional server-client operations, in choosing the best participants from all clients is still under-examined.

In this paper, we propose a novel communication-efficient FL framework by *jointly optimizing* momentum-based update and client selection, which can achieve better convergence rate and model accuracy, especially under highly skewed data distributions. To tackle the client drift issue, we first develop a *double* momentum SGD, where two customized momentum buffers are maintained on server and client sides so as to smooth global and local noisy update directions, respectively. In particular, our double momentum solution can be applied to a *wide range* of sampling schemes as long as they remain unbiased. Taking momentum results as input, we then devise an *adaptive* client selection algorithm to provide a proper representation for each client during FL aggregation. The key insight is that round-by-round model synchronizations require to dynamically sample the best participants that will facilitate higher learning performance. In general, our selection scheme can ensure *unbiased* aggregation without additional costly server-client communications and computations. In a nutshell, by tightly *coupling* the selection process with momentum design, we are able to effectively reduce the intrinsic variance induced by stochastic model update and client sampling. We also validate the efficiency of our approach both theoretically and empirically. The main contributions are summarized.

- We propose FedMoS, a communication-efficient FL framework with joint double momentum and adaptive selection. To our best knowledge, this is the first attempt that highlights the *inter-dependence* between momentum SGD and client selection to tame the client drift, thus navigating an *effective co-design* for promoting FL convergence and accuracy.
- FedMoS coordinates *customized* momentum buffers on the server and client sides to track global and local update directions, which greatly reduces the intrinsic model variance. Through a rigid analysis, we show that an *optimal* $O(T^{-2/3})$ *convergence rate* is achieved under *partial* client participation and *arbitrary* aggregation weight, which is also *orderly faster* than single momentum or plain gradient-based FL algorithms. Meanwhile, FedMoS attains favorable empirical performance even under highly skewed data distributions.
- Based on momentum characterization, FedMoS employs an *adaptive* selection scheme to cherry-pick representative clients for sampling variance reduction. Through *optimally* calibrating the sampling probabilities, our *unbiased* selection can substantially improve the communication efficiency *without inducing* extra server-client operation costs.
- Extensive experiments on *real-world datasets* corroborate the superiority of FedMoS over the state-of-the-art approaches. In particular, we can achieve the highest test accuracy with *2.4-7.5 times faster* convergence than benchmarks.

II. Preliminary and System Overview

We start by summarizing the basics of FL, and then present a high-level overview of FedMoS.

A. Federated Learning

Consider the cross-device FL system with total N clients $\mathcal{N} = \{1, 2, ..., N\}$ and a central *parameter server* (PS). Each client i maintains a local dataset $\mathcal{D}_i$ containing $D_i = |\mathcal{D}_i|$ data samples. Define $f_i(\boldsymbol{x}, \xi_i)$ as the loss function which measures the learning performance of model parameter $\boldsymbol{x}$ under input training sample $\xi_i \in \mathcal{D}_i$. On this basis, the loss of client i is

$$f_i(\boldsymbol{x}) = \frac{1}{D_i} \sum\nolimits_{\xi_i \in \mathcal{D}_i} f_i(\boldsymbol{x}, \xi_i). \tag{1}$$

Under the orchestration of PS, clients will collaboratively train a machine learning model without disclosing their raw data:

$$f(\boldsymbol{x}) \triangleq \sum\nolimits_{i \in \mathcal{N}} p_i f_i(\boldsymbol{x}), \tag{2}$$

where p_i signifies the importance of client i with $\sum_{i \in \mathcal{N}} p_i = 1$, and often $p_i = \frac{D_i}{\sum_{i \in \mathcal{N}} D_i}$. The objective is to find the optimal parameter for minimizing loss function, i.e., $f^* \triangleq \min_x f(\boldsymbol{x})$.

To this end, typical FL algorithms allow clients to perform multiple local SGD updates before a periodical global synchronization by PS [5], yet communication often becomes the primary bottleneck. To reduce the communication amount, we are faced with two non-trivial challenges. First, local datasets $\mathcal{D}_i, \forall i \in \mathcal{N}$ are distributed in a *non-i.i.d.* and *unbalanced* fashion across all clients, which introduces large variance and

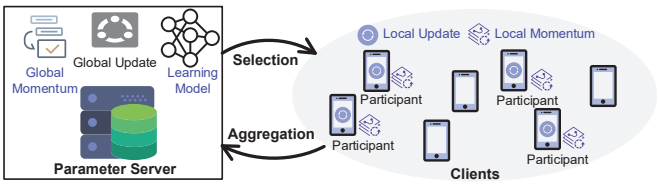


Fig. 1: A snapshot of FedMoS architecture.

discrepancy in their local model updates, namely *client drift*, so that the training process will be greatly prolonged. Second, selecting a portion of clients participating in each FL round can lead to a communication decline, but the incurred sampling variance may otherwise deteriorate the learning performance.

B. FedMoS Overview

We propose FedMoS by jointly designing double momentum SGD and adaptive client selection to handle the mentioned challenges. Fig. 1 shows the main architecture, where FedMoS proceeds in rounds of communication as described below.

- At the beginning of each round t, PS *adaptively* selects M out of N clients as participants, denoted by $\mathcal{S}^t$ with $|\mathcal{S}^t| = M$, and broadcasts the current global model $\boldsymbol{x}^t$ to $\mathcal{S}^t$;
- Any selected client $i \in \mathcal{S}^t$ initializes its local model $\boldsymbol{x}_{i,0}^t = \boldsymbol{x}^t$, then separately conducts I steps of *momentum SGD* on Eq. (1), and finally sends the updated model $\boldsymbol{x}_{i,I}^t$ to PS;
- PS aggregates the received parameters and performs momentum-based update to build a new global model $\boldsymbol{x}^{t+1}$.

1) Double Momentum: FedMoS customizes two momentum buffers for the client and server separately considering both the *coordination and difference* between local and global model updates. Due to skewed data distributions, reducing the inherent variance in *multi-step* local updates is our top priority. Hence, we propose a new variance reduction momentum $\boldsymbol{d}_{i,\tau}^t$ on client side to track the local update direction [15]:

$$\boldsymbol{d}_{i,\tau}^t = \tilde{\nabla}_{\mathcal{B}_{i,\tau}} f_i(\boldsymbol{x}_{i,\tau}^t) + (1-a)\big(\boldsymbol{d}_{i,\tau-1}^t - \tilde{\nabla}_{\mathcal{B}_{i,\tau}} f_i(\boldsymbol{x}_{i,\tau-1}^t)\big), \quad (3)$$

where $\tilde{\nabla}_{\mathcal{B}_{i,\tau}} f_i(\boldsymbol{x}_{i,\tau}^t) \triangleq \frac{1}{B}\sum_{\xi_i \in \mathcal{B}_{i,\tau}} \nabla f_i(\boldsymbol{x}_{i,\tau}^t, \xi_i)$ denotes the stochastic gradient on batch $\mathcal{B}_{i,\tau}$ in τ-th step. Also, a proximal term $\mu(\boldsymbol{x}_{i,\tau}^t - \boldsymbol{x}^t)$ is fused to further smooth out noise in local stochastic updates. Here, a and μ are constant coefficients.

On the other hand, PS mainly conducts an *one-step* global model update. This motivates us to maintain a Polyak momentum $\boldsymbol{u}^t$ [16] on the server side, which is effective when coordinating with the aggregated local updates:

$$\boldsymbol{u}^t = \beta \boldsymbol{u}^{t-1} - \frac{1}{\eta I}\sum\nolimits_{i\in\mathcal{S}^t} w_i^t(\boldsymbol{x}_{i,I}^t - \boldsymbol{x}^t), \quad (4)$$

in which w_i^t is the aggregation weight decided by client selection, β is a coefficient and η implies the training stepsize.

2) Client Selection: In principle, *unbiased* client sampling is the common consensus for improving FL performance [17]. A selection is regarded to be unbiased if the expectation of aggregated global model is equivalent to that of full client participation, i.e., $\mathbb{E}_{\mathcal{S}^t}[w_i^t] = p_i, \forall i \in \mathcal{N}$.

We propose an adaptive selection scheme that generalizes *multinomial distribution* (MD) sampling [17] where PS independently selects a client with support $\mathcal{N}$ for M times *with replacement*, i.e., $\mathcal{S}^t$ is a *multiset* where a client may appear more than once. Concretely, denote l_m as the sampled client in the m-th selection which follows MD such that client i is picked with probability $p_{i,m}$. For unbiasedness, w_i^t should be:

$$w_i^t = \frac{1}{M}\sum\nolimits_{m=1}^{M} \mathbb{I}(l_m = i). \quad (5)$$

Traditional MD sampling assumes a uniform sampling probability for each client across M selections, i.e., $\Pr(l_m = i) = p_i, \forall m = 1, \cdots, M$, which is inflexible and far from optimal. Considering this, FedMoS specifies MD sampling by picking $\mathcal{S}^t$ according to M *independent* distributions $\{p_{i,m} | i \in \mathcal{N}\}_{m=1}^M$, i.e., $\Pr(l_m = i) = p_{i,m}$ is *heterogeneous*. Coupled with the momentum-based update, our selection scheme seeks to *minimize the sampling variance and maintain unbiased aggregation* by determining the probabilities $\{p_{i,m} | i \in \mathcal{N}\}_{m=1}^M$.

Since the formal formulation of adaptive client selection tightly depends on the momentum result, we will defer and elaborate it in Section IV.

Algorithm 1: FedMoS: Double Momentum-Based Update with Adaptive Selection

Input: Coefficients a, η, μ, β, selected client size M, sampled data size B, local steps I

1 Initialize global momentum $\boldsymbol{u}^{-1}$ and model $\boldsymbol{x}^0$;
2 **for** $t = 0, \cdots, T-1$ **do**
3 PS selects M clients $\mathcal{S}^t$ based on adaptive client selection in Algorithm 2, then broadcasts $\boldsymbol{x}^t$;
4 **for** *client* $i \in \mathcal{S}^t$ **do**
5 **for** $\tau = 0, \cdots, I-1$ **do**
6 **if** $\tau = 0$ **then**
7 Initialize local model $\boldsymbol{x}_{i,\tau}^t = \boldsymbol{x}^t$;
8 Set local momentum $\boldsymbol{d}_{i,\tau}^t = \nabla f_i(\boldsymbol{x}_{i,\tau}^t)$;
9 **else**
10 Randomly sample a mini-batch $\mathcal{B}_{i,\tau}$ of data with size B from $\mathcal{D}_i$;
11 Compute $\boldsymbol{d}_{i,\tau}^t$ based on Eq. (3);
12 $\boldsymbol{x}_{i,\tau+1}^t = \boldsymbol{x}_{i,\tau}^t - \eta \boldsymbol{d}_{i,\tau}^t - \mu(\boldsymbol{x}_{i,\tau}^t - \boldsymbol{x}^t)$;
13 Send $(\boldsymbol{x}_{i,I}^t - \boldsymbol{x}^t)$ to PS;
  // Aggregation and momentum SGD at PS
14 Compute $\boldsymbol{u}^t$ from Eq. (4), $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \eta I \boldsymbol{u}^t$;

III. Double Momentum-Based FedMoS

In this section, we will illustrate the design of FedMoS.

A. FedMoS Design

Considering single momentum on either server or client side may be insufficient when data distributions diverge largely, we propose FedMoS to employ *double momentum buffers* on both sides instead. The pseudo-code is presented in Algorithm 1.

Concretely, we implement an unbiased and adaptive selection (Line 3) in Algorithm 2 to choose clients $\mathcal{S}^t$, which will be explored in later section. For each selected client, his local

model updates involve I steps of momentum-based computation (Line 12), where the stochastic gradient $\tilde{\nabla}_{\mathcal{B}_{i,\tau}} f_i(\boldsymbol{x}_{i,\tau}^t)$ is attained on the sampled batch $\mathcal{B}_{i,\tau}$ of data size B in all steps *excluding* the first step $\tau = 0$ to *remove the initial error* (Lines 6-11). The *local momentum* $\boldsymbol{d}_{i,\tau}^t$ is instantiated to be the gradient (Line 7), then renewed using $\tilde{\nabla}_{\mathcal{B}_{i,\tau}} f_i(\boldsymbol{x}_{i,\tau}^t)$ and last-step value $\boldsymbol{d}_{i,\tau-1}^t$ based on Eq. (3). A proximal term $\mu(\boldsymbol{x}_{i,\tau}^t - \boldsymbol{x}^t)$ is also integrated to further smooth local model updates. On server side, upon receiving all parameters, *global momentum* $\boldsymbol{u}^t$ is recalculated, and then global model $\boldsymbol{x}^{t+1}$ is updated following the Polyak momentum SGD (Line 14).

Next, we analyze the convergence of FedMoS given unbiased sampling, i.e., not limited to the client selection implemented in Algorithm 2. Prior to the analysis, we first introduce several common assumptions to assist our later elaborations.

B. Model Assumptions

For analytical tractability, we state the following assumptions pertaining to the machine learning model, which have been made in a fair amount of previous works [18], [19].

Assumption 1 (Smoothness)**.** *For each client $i \in \mathcal{N}$, the stochastic loss $f_i(\boldsymbol{x}, \xi_i), \forall \xi_i \in \mathcal{D}_i$ is L-smooth, i.e., $\|\nabla f_i(\boldsymbol{x}, \xi_i) - \nabla f_i(\boldsymbol{y}, \xi_i)\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$ for all $\boldsymbol{x}$ and $\boldsymbol{y}$.*

Assumption 2 (Bounded Variance)**.** *For each client $i \in \mathcal{N}$, the variance of local stochastic gradient is bounded, i.e., $\mathbb{E}[\|\nabla f_i(\boldsymbol{x}, \xi_i) - \nabla f_i(\boldsymbol{x})\|^2] \leq \sigma^2, \forall \xi_i \in \mathcal{D}_i$.*

Assumption 3 (Bounded Dissimilarity)**.** *For each client $i \in \mathcal{N}$, the loss function satisfies $\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\| \leq \varsigma^2$.*

Assumption 4 (PL Condition)**.** *Loss function $f(\boldsymbol{x})$ satisfies ζ-Polyak-Łojasiewicz (PL) condition if $\|\nabla f(\boldsymbol{x})\|^2 \geq 2\zeta(f(\boldsymbol{x}) - f^*)$ where $f^* \triangleq \min_{\boldsymbol{x}} f(\boldsymbol{x})$.*

Assumption 1 also implies that the loss functions $f_i(\boldsymbol{x})$ and $f(\boldsymbol{x})$ are L-smooth. Assumption 2 quantifies the local stochastic variance which can be regarded as intra-client heterogeneity, while in contrast Assumption 3 measures the data heterogeneity across clients. PL condition in Assumption 4 is weaker than the ζ-strongly convexity which is required in many existing works [13], [20], and we will provide the convergence results with and without this condition, respectively.

C. Convergence Rate

1) Main Results: Given the model assumptions, we now show the main results of FedMoS, i.e., the convergence rate pertaining to total communication round T. For ease of exposition, denote $p_{\max} = \max_{i \in \mathcal{N}} p_i$ as the highest importance.

Theorem 1. *Suppose $\frac{L\eta}{\mu} \leq \frac{1}{158\sqrt{N\sum_{i\in\mathcal{N}} p_i^2}}$, $L\eta \leq \frac{M}{2048Np_{\max}}$, $\eta = \mathcal{O}(T^{-2/3}), \mu = \mathcal{O}(T^{-1/3}), a = \mathcal{O}(T^{-1/2})$, and $\mu I \in [1, \frac{9}{8}]$ with $\mu \leq \frac{1}{2}$. Under Assumptions 1-3, we have:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(\boldsymbol{x}^t)\|^2] \leq \mathcal{O}\Big(\frac{f(\boldsymbol{x}^0) - f^*}{T^{2/3}}\Big) + \mathcal{O}\Big(\frac{\sigma^2}{T^{2/3}}\Big) + \mathcal{O}\Big(\frac{\varsigma^2}{T^{2/3}}\Big). \tag{6}$$

Besides, under Assumptions 1-4, we attain:

$$\frac{1}{T}\sum_{t=0}^{T-1}(\mathbb{E}[f(\boldsymbol{x}^t)] - f^*) \leq \mathcal{O}\Big(\frac{f(\boldsymbol{x}^0) - f^*}{T^{2/3}\zeta}\Big) + \mathcal{O}\Big(\frac{\sigma^2}{T^{2/3}\zeta}\Big) + \mathcal{O}\Big(\frac{\varsigma^2}{T^{2/3}\zeta}\Big). \tag{7}$$

The proof is in [36] Appendix A1. Theorem 1 presents a unified $\mathcal{O}(T^{-2/3})$ convergence rate when PL condition holds or not. Comparing with extant convergence achievements, we have the following observations.

- Convergence rate of FedAvg [21] or FedProx [6] is $\mathcal{O}(T^{-1/2})$, i.e., it needs $\mathcal{O}(\epsilon^2)$ communication rounds to approach an ϵ-stationary point. Since FedMoS yields an $\mathcal{O}(T^{-2/3})$ convergence, communications required to reach an ϵ-stationary point are significantly reduced to $\mathcal{O}(\epsilon^{3/2})$, which has been proved to be *optimal for non-convex optimizations* [22] and is *orderly faster* than FedAvg or FedProx.
- Single momentum-based FL algorithms lead to a theoretical $\mathcal{O}(T^{-1/2})$ convergence rate [7], that is *slower* than FedMoS.
- FedLOMO [18] and STEM [19] attain a similar $\mathcal{O}(T^{-2/3})$ convergence result. However, they both consider a *uniform* importance, i.e., $p_i = \frac{1}{N}, \forall i \in \mathcal{N}$, which plays an important role in the convergence proof. Besides, STEM is built on *full* client participation. By incorporating the proximal term and arbitrary weight, FedMoS is more *flexible* and *general* even when facing practically skewed data distributions to achieve better performance via cherry-picking representative clients.

Remark: Theoretical $\mathcal{O}(T^{-2/3})$ holds as long as the client selection is unbiased, while its implementation, like Algorithm 2, mainly promotes the *empirical performance*.

2) FedMoS Extension: Previously, the convergence of FedMoS is obtained by instantiating local $\boldsymbol{d}_{i,0}^t$ to be the gradient $\nabla f_i(\boldsymbol{x}_{i,0}^t)$, i.e., initial error is $\mathbb{E}[\|\boldsymbol{d}_{i,0}^t - \nabla f_i(\boldsymbol{x}_{i,0}^t)\|^2] = 0$, and then using the stochastic gradient on a sampled mini-batch at the rest $I - 1$ steps. In fact, we can *unify* the gradient calculation throughout the whole I local updates, where the first sampled batch size is denoted as $B_{i,0}$ with *initial stochastic gradient* being $\tilde{\nabla}_{\mathcal{B}_{i,0}} f_i(\boldsymbol{x}_{i,0}^t)$ and the rest of steps remaining the same as in Algorithm 1. The corresponding convergence is derived below with the proof in [36] Appendix A2.

Corollary 1. *Set parameters as in Theorem 1, and suppose $B_{i,0} = \min\{\Omega(T^{2/3}), D_i\}, \forall i \in \mathcal{N}$. Under Assumptions 1-3, we attain:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(\boldsymbol{x}^t)\|^2] \leq \mathcal{O}\Big(\frac{f(\boldsymbol{x}^0) - f^*}{T^{2/3}}\Big) + \mathcal{O}\Big(\frac{\sigma^2}{T^{2/3}}\Big) + \mathcal{O}\Big(\sigma^2\sum_{i\in\mathcal{N}}\frac{1}{B_{i,0}}\Big) + \mathcal{O}\Big(\frac{\varsigma^2}{T^{2/3}}\Big). \tag{8}$$

Besides, under Assumptions 1-4, we have:

$$\frac{1}{T}\sum_{t=0}^{T-1}(\mathbb{E}[f(\boldsymbol{x}^t)] - f^*) \leq \mathcal{O}\Big(\frac{f(\boldsymbol{x}^0) - f^*}{T^{2/3}\zeta}\Big) + \mathcal{O}\Big(\frac{\sigma^2}{T^{2/3}\zeta}\Big) + \mathcal{O}\Big(\frac{\sigma^2}{\zeta}\sum_{i\in\mathcal{N}}\frac{1}{B_{i,0}}\Big) + \mathcal{O}\Big(\frac{\varsigma^2}{T^{2/3}\zeta}\Big). \tag{9}$$

From Eqs. (8)-(9), the convergence rate is still $\mathcal{O}(T^{-2/3})$ since the term $\mathcal{O}\big(\sigma^2\sum_{i\in\mathcal{N}}\frac{1}{B_{i,0}}\big)$ or $\mathcal{O}\big(\frac{\sigma^2}{\zeta}\sum_{i\in\mathcal{N}}\frac{1}{B_{i,0}}\big)$ will vanish if $\min\{\Omega(T^{2/3}), D_i\} = D_i$, which boils down to the original case of Theorem 1. Comparing Theorem 1 with Corollary 1, the main difference lies in the initial error, which is non-zero and altered by the first batch size $B_{i,0}$ for Corollary 1. Since this error will *propagate* along with local model updates, captured by $\mathcal{O}\big(\sigma^2\sum_{i\in\mathcal{N}}\frac{1}{B_{i,0}}\big)$ or $\mathcal{O}\big(\frac{\sigma^2}{\zeta}\sum_{i\in\mathcal{N}}\frac{1}{B_{i,0}}\big)$, we need to bound the corresponding term via adjusting $B_{i,0}$ so as to achieve the same order of convergence result.

D. Proof Outline

Due to space limit, we only provide the proof outline of Theorem 1, which is composed of a series of lemmas. Afterwards, the proof of Corollary 1 can be similarly characterized. According to Assumption 1 and FedMoS, we have:

$$\begin{aligned}&\mathbb{E}[f(\boldsymbol{x}^{t+1})] \le \mathbb{E}[f(\boldsymbol{x}^t)]\\&+\mathbb{E}[\langle\nabla f(\boldsymbol{x}^t), \boldsymbol{x}^{t+1}-\boldsymbol{x}^t\rangle] + \frac{L}{2}\mathbb{E}[\|\boldsymbol{x}^{t+1}-\boldsymbol{x}^t\|^2]\\&=\mathbb{E}[f(\boldsymbol{x}^t)] + \eta I\mathbb{E}[\langle\nabla f(\boldsymbol{x}^t), -\boldsymbol{u}^t\rangle] + \frac{L}{2}\eta^2I^2\mathbb{E}[\|\boldsymbol{u}^t\|^2].\end{aligned} \tag{10}$$

The convergence will be deduced by *centering around* this inequality. For convenience, we define the following notations:

$$\boldsymbol{e}_{i,\tau}^t \triangleq \boldsymbol{d}_{i,\tau}^t - \nabla f_i(\boldsymbol{x}_{i,\tau}^t), \tag{11}$$

$$\overline{\boldsymbol{d}}_\tau^t \triangleq \sum_{i\in\mathcal{N}} p_i\boldsymbol{d}_{i,\tau}^t,\ \tilde{\boldsymbol{d}}_\tau^t \triangleq \sum_{i\in\mathcal{S}^t} w_i^t\boldsymbol{d}_{i,\tau}^t. \tag{12}$$

Particularly, $\boldsymbol{e}_{i,\tau}^t$ embodies the error between local momentum and corresponding gradient in τ-th step, i.e., we have a dynamically changing error. Besides, $\overline{\boldsymbol{d}}_\tau^t$ and $\tilde{\boldsymbol{d}}_\tau^t$ indicate the expected and weighted aggregated momentum, respectively, signifying the momentum difference due to client selection.

Cross term. We first disassemble the cross term, namely the second part in Eq. (10). All proofs are in [36] Appendix B.

Lemma 1. *According to Algorithm 1, we have:*

$$\begin{aligned}&\mathbb{E}[\langle\nabla f(\boldsymbol{x}^t), -\boldsymbol{u}^t\rangle] = \mathbb{E}[\langle\nabla f(\boldsymbol{x}^t), -\beta\boldsymbol{u}^{t-1}\rangle]\\&+\mathbb{E}\Big[\Big\langle\nabla f(\boldsymbol{x}^t), -\frac{1}{I}\sum_{\tau=0}^{I-1}(1-\mu)^{I-1-\tau}\tilde{\boldsymbol{d}}_\tau^t\Big\rangle\Big].\end{aligned} \tag{13}$$

This lemma is derived by applying the global model aggregation and server momentum update based on Eq. (4), which is the cornerstone for convergence analysis. We continue to handle the last part in Eq. (13).

Lemma 2. *From momentum-based FedMoS, we attain:*

$$\begin{aligned}&\sum_{\tau=0}^{I-1}(1-\mu)^{I-1-\tau}\mathbb{E}[\langle\nabla f(\boldsymbol{x}^t), -\tilde{\boldsymbol{d}}_\tau^t\rangle]\\&\le\sum_{\tau=0}^{I-1}(1-\mu)^{I-1-\tau}\Big(-\frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{x}^t)\|^2] - \frac{1}{2}\mathbb{E}[\|\overline{\boldsymbol{d}}_\tau^t\|^2]\\&+\frac{L^2\eta^2\sum_{i\in\mathcal{N}}p_i^2}{3\mu}\sum_{i\in\mathcal{N}}\sum_{k=0}^{\tau-1}\mathbb{E}[\|\boldsymbol{d}_{i,k}^t\|^2] + \frac{\sum_{i\in\mathcal{N}}p_i^2}{2}\sum_{i\in\mathcal{N}}\mathbb{E}[\|\boldsymbol{e}_{i,\tau}^t\|^2]\Big).\end{aligned} \tag{14}$$

From Lemma 2, one can characterize Eq. (13) via analyzing each constituent in Eq. (14), mainly the momentum and error.

Local momentum and error. Based on Lemmas 1 and 2, we next discuss the local momentum $\mathbb{E}[\|\boldsymbol{d}_{i,\tau}^t\|^2]$ and error $\mathbb{E}[\|\boldsymbol{e}_{i,\tau}^t\|^2]$ since they both are involved in Eq. (14). Please refer to Appendix C in [36] for the proofs.

Lemma 3. *According to Assumption 1, and assuming* $\frac{L\eta}{\mu} \le \frac{1}{158\sqrt{N\sum_{i\in\mathcal{N}}p_i^2}}, \mu I \le \frac{9}{8}, \mu \le \frac{1}{2}$*, the momentum* $\boldsymbol{d}_{i,\tau}^t$ *satisfies:*

$$\sum_{\tau=0}^{I-1}\mathbb{E}[\|\boldsymbol{d}_{i,\tau}^t\|^2] \le \frac{256}{3\mu}\mathbb{E}[\|\nabla f_i(\boldsymbol{x}^t)\|^2] + \frac{16a^2I^2}{B}\sigma^2. \tag{15}$$

As for the error $\mathbb{E}[\|\boldsymbol{e}_{i,\tau}^t\|^2]$, the following conclusion holds.

Lemma 4. *Based on Assumption 2, the error* $\boldsymbol{e}_{i,\tau}^t$ *satisfies:*

$$\begin{aligned}&\sum_{\tau=0}^{I-1}(1-\mu)^{I-1-\tau}\mathbb{E}[\|\boldsymbol{e}_{i,\tau}^t\|^2]\\&\le\frac{2a^2I}{B\mu}\sigma^2 + \frac{2L^2}{\mu}\sum_{\tau=0}^{I-1}\mathbb{E}[\|\boldsymbol{x}_{i,\tau+1}^t - \boldsymbol{x}_{i,\tau}^t\|^2].\end{aligned} \tag{16}$$

To obtain the bound of $\mathbb{E}[\|\boldsymbol{e}_{i,\tau}^t\|^2]$ in Lemma 4, it is critical to cap the term $\mathbb{E}[\|\boldsymbol{x}_{i,\tau}^t - \boldsymbol{x}_{i,\tau-1}^t\|^2]$ which is shown below.

Lemma 5. *From Lemma 3, we obtain:*

$$\sum_{\tau=0}^{I-1}\mathbb{E}[\|\boldsymbol{x}_{i,\tau+1}^t - \boldsymbol{x}_{i,\tau}^t\|^2] \le \tfrac{128\eta^2I}{\mu}\mathbb{E}[\|\nabla f_i(\boldsymbol{x}^t)\|^2] + \tfrac{24a^2I^3\eta^2}{B}\sigma^2. \tag{17}$$

Lemmas 1-5 enable bounding the cross term in Eq. (10). The remaining issue is to deal with global momentum $\mathbb{E}[\|\boldsymbol{u}^t\|^2]$.

Global momentum. Momentum $\boldsymbol{u}^t$ depends on both its previous buffer and current global synchronization as in Eq. (4). Considering that Eq. (14) contains the expected momentum $\overline{\boldsymbol{d}}_\tau^t$, we should expand $\boldsymbol{u}^t$ in terms of $\overline{\boldsymbol{d}}_\tau^t$ to merge this term. Proofs are presented in [36] Appendix D.

Lemma 6. *From Assumption 2 and Lemma 3, we have:*

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{u}^t\|^2] \le 2\beta^2\mathbb{E}[\|\boldsymbol{u}^{t-1}\|^2] + \frac{2p_{\max}}{MI^2}\sum_{i\in\mathcal{N}}\Big(\frac{256}{3\mu}\mathbb{E}[\|\nabla f_i(\boldsymbol{x}^t)\|^2]\\+\frac{16a^2I^2}{B}\sigma^2\Big) + \frac{4}{3I}\sum_{\tau=0}^{I-1}\mathbb{E}[\|\overline{\boldsymbol{d}}_\tau^t\|^2].\end{aligned} \tag{18}$$

Using Lemmas 1-6, we provide a close result to Theorem 1.

Lemma 7. *Under the conditions in Theorem 1, we obtain:*

$$\begin{aligned}&\mathbb{E}[f(\boldsymbol{x}^{t+1})] \le \mathbb{E}[f(\boldsymbol{x}^t)] - \frac{\eta}{8\mu}\mathbb{E}[\|\nabla f(\boldsymbol{x}^t)\|^2] + C\sum_{i\in\mathcal{N}}\mathbb{E}[\|\nabla f_i(\boldsymbol{x}^t)\|^2]\\&+C'\sigma^2 + \Big(2\eta\mu\beta^2I^2 + \frac{4L\eta^2\beta^2I^2}{3}\Big)\mathbb{E}[\|\boldsymbol{u}^{t-1}\|^2] - \frac{L\eta^2I^2}{6}\mathbb{E}[\|\boldsymbol{u}^t\|^2],\end{aligned} \tag{19}$$

where $C = (\frac{256}{9}+144)\frac{L^2\eta^3\sum_{i\in\mathcal{N}}p_i^2}{\mu^3} + \frac{1024L\eta^2p_{\max}}{9\mu M}$ *and* $C' = \frac{\eta a^2IN\sum_{i\in\mathcal{N}}p_i^2}{\mu B} + \frac{16L^2\eta^3a^2I^2N\sum_{i\in\mathcal{N}}p_i^2}{3\mu^2B} + \frac{24L^2\eta^3a^2I^3N\sum_{i\in\mathcal{N}}p_i^2}{\mu B} + \frac{64L\eta^2a^2I^2Np_{\max}}{3MB}$.

In the light of Lemma 7, we can apply a telescope sum on both sides of Eq. (19) from 0 to T and combine Assumptions 3-4 to derive the convergence rate in Theorem 1. As for

Corollary 1, the convergence is obtained by further characterizing the propagation of initial error $\mathbb{E}[\|\boldsymbol{d}_{i,0}^t - \nabla f_i(\boldsymbol{x}_{i,0}^t)\|^2]$, especially adjusting the error in Lemma 4 and its successor results. This will additionally introduce terms $\frac{144\mu}{\eta}C_0\sigma^2$ and $\frac{72\mu}{\eta\zeta}C_0\sigma^2$ corresponding to $\mathcal{O}\big(\sigma^2\sum_{i\in\mathcal{N}}\frac{1}{B_{i,0}}\big)$ in Eq. (8) and $\mathcal{O}\big(\frac{\sigma^2}{\zeta}\sum_{i\in\mathcal{N}}\frac{1}{B_{i,0}}\big)$ in Eq. (9), respectively, where $C_0 = \frac{\eta N\sum_{i\in\mathcal{N}}\frac{p_i^2}{B_{i,0}}}{2\mu} + \frac{16L^2\eta^3IN\sum_{i\in\mathcal{N}}\frac{p_i^2}{B_{i,0}}}{3\mu^2} + \frac{24L^2\eta^3I^2N\sum_{i\in\mathcal{N}}\frac{p_i^2}{B_{i,0}}}{\mu} + \sum_{i\in\mathcal{N}}\frac{64L\eta^2Ip_{\max}}{3MB_{i,0}}$. Overall, we complete the proof outline. Proof details are omitted in this paper owing to limited space.

IV. Adaptive Client Selection Scheme

Double momentum can promote FL performance by mitigating the variance inherent in stochastic updates. But client selection would incur *sampling variance*, nonetheless. In this section, we elaborate the adaptive selection scheme of FedMoS to cherry-pick best clients for sampling variance reduction.

A. Unbiased Client Selection

FedMoS in Algorithm 1 implies $\boldsymbol{u}^t = \beta\boldsymbol{u}^{t-1} - \frac{1}{\eta I}\sum_{i\in\mathcal{S}^t} w_i^t(\boldsymbol{x}_{i,I}^t - \boldsymbol{x}^t)$. Also, from Lemma 1, we have:

$$\begin{aligned}\sum_{i\in\mathcal{S}^t} w_i^t(\boldsymbol{x}_{i,I}^t - \boldsymbol{x}^t) &= \eta\sum_{\tau=0}^{I-1}(1-\mu)^{I-1-\tau}\sum_{i\in\mathcal{S}^t} w_i^t\boldsymbol{d}_{i,\tau}^t \\ &= \eta\sum_{\tau=0}^{I-1}(1-\mu)^{I-1-\tau}\sum_{i\in\mathcal{N}} w_i^t\boldsymbol{d}_{i,\tau}^t,\end{aligned} \tag{20}$$

where the last equality is because $w_i^t = 0$ based on Eq. (5) when client i is not selected. Therefore, FedMoS, including the client selection, is unbiased if the following condition holds:

$$\mathbb{E}\Big[\sum_{\tau=0}^{I-1}(1-\mu)^{I-1-\tau}\sum_{i\in\mathcal{N}} w_i^t\boldsymbol{d}_{i,\tau}^t\Big] = \sum_{\tau=0}^{I-1}(1-\mu)^{I-1-\tau}\sum_{i\in\mathcal{N}} p_i\nabla f_i(\boldsymbol{x}_{i,\tau}^t). \tag{21}$$

We first show that local momentum is an unbiased estimation of the gradient value, where proof is in [36] Appendix E.

Lemma 8. *For any client $i\in\mathcal{N}$, the local momentum satisfies $\mathbb{E}_{\prod_{k=0}^{\tau}\mathcal{B}_{i,k}}[\boldsymbol{d}_{i,\tau}^t] = \nabla f_i(\boldsymbol{x}_{i,\tau}^t)$ based on Algorithm 1.*

Lemma 8 implies that the unbiased FedMoS needs to ensure unbiased sampling, i.e., $\mathbb{E}_{\mathcal{S}^t}[w_i^t] = p_i$. Since w_i^t is decided by the client selection in Eq. (5) where $w_i^t = \frac{1}{M}\sum_{m=1}^M\mathbb{I}(l_m = i)$ with $\Pr(l_m = i) = p_{i,m}$, we have $\mathbb{E}_{\mathcal{S}^t}[w_i^t] = \frac{1}{M}\sum_{m=1}^M p_{i,m}$. As a result, the *unbiased sampling* indeed requires:

$$\sum_{m=1}^M p_{i,m} = Mp_i, \forall i\in\mathcal{N}. \tag{22}$$

In a nutshell, we will design an adaptive client selection to reduce the sampling variance while also maintaining unbiased.

B. Selection Problem Formulation

Achieving unbiasedness and variance reduction is critical to advocating FL performance [17]. Using Eq. (20), we compute the variance of stochastic update in t-th round as $\mathrm{var} \triangleq \mathbb{E}[\|\sum_{\tau=0}^{I-1}\eta(1-\mu)^{I-1-\tau}\sum_{i\in\mathcal{N}} w_i^t\boldsymbol{d}_{i,\tau}^t - \sum_{\tau=0}^{I-1}\eta(1-\mu)^{I-1-\tau}\sum_{i\in\mathcal{N}} p_i\nabla f(\boldsymbol{x}^t)\|^2]$, which is further decomposed:

$$\begin{aligned}\mathrm{var} &= \underbrace{\mathbb{E}\Big[\Big\|\sum_{\tau=0}^{I-1}\eta(1-\mu)^{I-1-\tau}\sum_{i\in\mathcal{N}}(w_i^t\boldsymbol{d}_{i,\tau}^t - p_i\nabla f_i(\boldsymbol{x}_{i,\tau}^t))\Big\|^2\Big]}_{\mathrm{var1}} \\ &+ \mathbb{E}\Big[\Big\|\sum_{\tau=0}^{I-1}\eta(1-\mu)^{I-1-\tau}\sum_{i\in\mathcal{N}} p_i(\nabla f_i(\boldsymbol{x}_{i,\tau}^t) - f(\boldsymbol{x}^t))\Big\|^2\Big].\end{aligned} \tag{23}$$

Eq. (23) holds because the cross term is 0 from Lemma 8. Apparently, only the first part var1 is altered by client selection while the rest relies on the model update of each client. Then,

$$\begin{aligned}\mathrm{var1} &= \eta^2\sum_{i\in\mathcal{N}}\sum_{\tau=0}^{I-1}(1-\mu)^{2I-2-2\tau}\mathbb{E}[\|w_i^t\boldsymbol{d}_{i,\tau}^t - p_i\nabla f_i(\boldsymbol{x}_{i,\tau}^t)\|^2] \\ &= \eta^2\sum_{i\in\mathcal{N}}\sum_{\tau=0}^{I-1}(1-\mu)^{2I-2-2\tau}\mathbb{E}[\|w_i^t\boldsymbol{d}_{i,\tau}^t\|^2 - \|p_i\nabla f_i(\boldsymbol{x}_{i,\tau}^t)\|^2],\end{aligned} \tag{24}$$

in which the last equality is due to $\mathbb{E}[w_i^t\boldsymbol{d}_{i,\tau}^t] = p_i\nabla f_i(\boldsymbol{x}_{i,\tau}^t)$ since FedMoS is unbiased. An important observation is that merely the first term of Eq. (24), involving $\mathbb{E}[\|w_i^t\boldsymbol{d}_{i,\tau}^t\|^2] = \sum_{i\in\mathcal{N}}\sum_{\tau=0}^{I-1}(1-\mu)^{2I-2-2\tau}\mathbb{E}_{\mathcal{S}^t}[(w_i^t)^2]\mathbb{E}[\|\boldsymbol{d}_{i,\tau}^t\|^2]$, depends on the selection. Along with unbiased MD sampling in Eq. (22), we express the aggregation weight $\mathbb{E}_{\mathcal{S}^t}[(w_i^t)^2]$ as:

$$\begin{aligned}\mathbb{E}_{\mathcal{S}^t}[(w_i^t)^2] &= \mathbb{E}_{\mathcal{S}^t}\Big[\Big(\frac{1}{M}\sum\nolimits_{m=1}^M\mathbb{I}(l_m = i)\Big)^2\Big] \\ &= \frac{1}{M^2}\Big(Mp_i + M^2p_i^2 - \sum\nolimits_{m=1}^M p_{i,m}^2\Big).\end{aligned} \tag{25}$$

Given the fact that p_i is prior information and $\boldsymbol{d}_{i,\tau}^t$ is influenced by local model updates, we ought to maximize $\sum_{i\in\mathcal{N}}\sum_{\tau=0}^{I-1}(1-\mu)^{2I-2-2\tau}\mathbb{E}[\|\boldsymbol{d}_{i,\tau}^t\|^2]\sum_{m=1}^M p_{i,m}^2$ to reduce the variance in line with Eqs. (23)-(25).

However, it is challenging to accomplish this goal since each $\boldsymbol{d}_{i,\tau}^t$ is *undisclosed* at the time of client selection. To overcome the problem, we divide the objective and seek to optimize $\sum_{\tau=0}^{I-1}(1-\mu)^{2I-2-2\tau}\mathbb{E}[\|\boldsymbol{d}_{i,\tau}^t\|^2]\sum_{m=1}^M p_{i,m}^2$ for each individual client due to their *independent local updates*. By doing this, it dispenses us with the burden of considering $\boldsymbol{d}_{i,\tau}^t$, so we can focus on maximizing $\sum_{m=1}^M p_{i,m}^2$ instead. In general, we determine the sampling probabilities $\{p_{i,m}|i\in\mathcal{N}\}_{m=1}^M$ via solving the *client selection problem* below:

$$\begin{aligned}\max\quad & \sum_{m=1}^M p_{i,m}^2 \\ \text{s.t.}\quad & \sum_{m=1}^M p_{i,m} = Mp_i, \forall i\in\mathcal{N} \\ & \sum_{i\in\mathcal{N}} p_{i,m} = 1, \forall m = 1,\cdots,M.\end{aligned} \tag{26}$$

The first constraint is simply the unbiased requirement, and the second is raised by MD sampling where the total probability in any selection should be 1. One can see that the selection problem corresponding to each client is inter-dependent, which calls for a holistic sampling scheme design.

Algorithm 2: Adaptive Client Selection

Input: Importance $\{p_i\}$, selected client size M

1 Sort clients by descending order of importance $\{p_i\}$;
2 Initialize $P_i = 0, \forall i \in \mathcal{N}$;
3 **for** $m = 1, \cdots, M$ **do**
4 $\text{sum} = 0$, $p_{i,m} = 0, \forall i \in \mathcal{N}$;
5 **for** *client* $i \in \mathcal{N}$ **do**
6 **if** $P_i < Mp_i$ **then**
7 $p_{i,m} = \min\{Mp_i - P_i, 1\}$;
8 $\text{sum1} = \text{sum}$, $\text{sum} = \text{sum1} + p_{i,m}$;
9 **if** $\text{sum} \leq 1$ **then**
10 $P_i = P_i + p_{i,m}$;
11 **else**
12 $p_{i,m} = 1 - \text{sum1}$, $P_i = P_i + p_{i,m}$, **break**;
13 Select a client based on MD sampling according to probabilities $\{p_{i,m} | i \in \mathcal{N}\}$;

C. Adaptive Selection Scheme Design

We have two observations for Eq. (26). First, the constraints are *not linearly independent* as $\sum_{i\in\mathcal{N}} p_i = 1$. Second, there are NM variables, which are far more than $(N + M)$ constraints, i.e., the *solution space* is very large. Therefore, we propose an adaptive selection scheme to solve Eq. (26) by optimizing the sampling probability for each client *successively*.

1) Adaptive Client Selection: Recall that our goal is to maximize $\sum_{m=1}^{M} p_{i,m}^2$ under the unbiased and MD constraints. Based on Cauchy–Schwarz inequality, we have:

$$\sum\nolimits_{m=1}^{M} p_{i,m}^2 \geq \frac{1}{M}\Big(\sum\nolimits_{m=1}^{M} p_{i,m}\Big)^2 = Mp_i^2, \qquad (27)$$

where $=$ holds only when $p_{i,m} = p_i, \forall m$, i.e., setting a uniform probability as in MD sampling leads to the worst performance. In principle, one should assign *uneven probabilities* across M times of sampling to solve Eq. (26). Obeying this rule, Algorithm 2 presents the adaptive client selection.

The philosophy behind Algorithm 2 is simple yet effective, that is handling each client *sequentially* and then *concentrating* his sampling probability on one or several *consecutive* times in order to optimize $\sum_{m=1}^{M} p_{i,m}^2$ (Lines 5-12). One distinct difference of our selection from existing schemes [11], [13], [14], [23] is that we require no prior client information, which eliminates the costly server-client communications. More importantly, this also avoids all clients from intensive gradient computation. Note that clients will be sorted in a descending order of their importance (Line 1). The sorting operation, with computation complexity of $\mathcal{O}(N \log N)$, enables sampling higher-importance clients in the first several times to prevent their total weights Mp_i from being divided into multiple fragmented parts. Next, we show that Algorithm 2 is unbiased, and the proof is presented in [36] Appendix F.

Theorem 2. *Algorithm 2 outputs an unbiased client selection, where the expectation of weight satisfies* $\mathbb{E}[w_i^t] = p_i, \forall i \in \mathcal{N}$.

Remark: General FL algorithms are implemented atop of *uniform client sampling* (UCS) [5], [6], which selects clients uniformly at random with probability p_i and sets weight as:

$$w_i^t = \mathbb{I}(i \in \mathcal{S}^t)\frac{N}{M}p_i. \qquad (28)$$

We explain why *adaptive client selection* (ACS) is superior to UCS. Analogous to Eq. (25), the sampling variance of client i is $\text{var}_i^{\text{ACS}} = \mathbb{E}_{\mathcal{S}^t}[(w_i^t)^2] - p_i^2 = \frac{1}{M}p_i - \frac{1}{M^2}\sum_{m=1}^{M} p_{i,m}^2$. Roughly speaking, any client i will be selected with probability 1 for about Mp_i times when $Mp_i \geq 1$, or with probability Mp_i for one time. Then, Algorithm 2 yields $\text{var}_i^{\text{ACS}} \approx 0$ if $Mp_i \geq 1$ or $\text{var}_i^{\text{ACS}} = \frac{1}{M}p_i - p_i^2$ otherwise. Regarding uniform sampling, the variance is $\text{var}_i^{\text{UCS}} = \mathbb{E}_{\mathcal{S}^t}[(w_i^t)^2] - p_i^2 = \frac{N}{M}p_i^2 - p_i^2$ based on Eq. (28). For $Mp_i \geq 1$, obviously $\text{var}_i^{\text{UCS}} \geq \text{var}_i^{\text{ACS}}$ holds, and for $Mp_i < 1$, we still have $\text{var}_i^{\text{UCS}} \approx \text{var}_i^{\text{ACS}}$ since $Np_i \sim 1$. Therefore, the variance of Algorithm 2 will be much smaller.

2) Clustered Adaptive Client Selection: Without collecting any information of clients, like the gradient, Algorithm 2 reduces the variance by unevenly distributing sampling probabilities over M selections, which releases them from expensive communications and computations. To facilitate better client representativity, we next characterize a variant of Algorithm 2, i.e., clustered sampling [10], where a warm-up phase is required to obtain the client "similarity".

Generally, the similarity measures local data disparity, which is used to pick *congruent data* in training *representation*, thereby leading to better FL convergence and accuracy. For privacy preservation, clients only upload model parameters rather than raw data. Therefore, we leverage T_e extra communication rounds to acquire each gradient $\nabla f_i(\boldsymbol{x}^{T_e}), \forall i \in \mathcal{N}$ so as to quantify the similarity and bypass the privacy concern [11], [14]. On the top of Algorithm 2, we elucidate the main idea of the clustered adaptive scheme below.

- All clients and PS run T_e-round FedMoS to learn $\nabla f_i(\boldsymbol{x}^{T_e})$;
- Calculate similarity between clients based on cosine distance, i.e., $s_{i,j} = \frac{\langle \nabla f_i(\boldsymbol{x}^{T_e}), \nabla f_j(\boldsymbol{x}^{T_e})\rangle}{\|\nabla f_i(\boldsymbol{x}^{T_e})\| \|\nabla f_j(\boldsymbol{x}^{T_e})\|}$;
- Cluster clients into disjoint sets $\mathcal{C} = \{c_1, c_2, \cdots, c_K\}$ using $s_{i,j}$, with "close" ones being placed into the same set;
- Sort sets by descending order of total importance $\sum_{i\in c_k} p_i$;
- Execute the same process as Algorithm 2 to calculate the sampling probability and select clients accordingly.

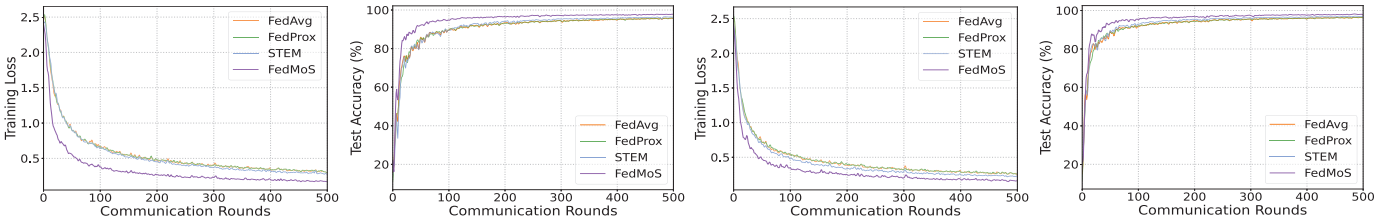
Consistent with Theorem 2, the clustered adaptive selection is also unbiased. Overall, our adaptive selection schemes solve Eq. (26) with a low computation complexity, since we will assign client sampling probability to an extreme value, mostly 0 or 1, in a sequential manner for maximizing the objective.

V. Performance Evaluation

In this section, we evaluate the performance of FedMoS, including momentum-based update and client selection scheme.

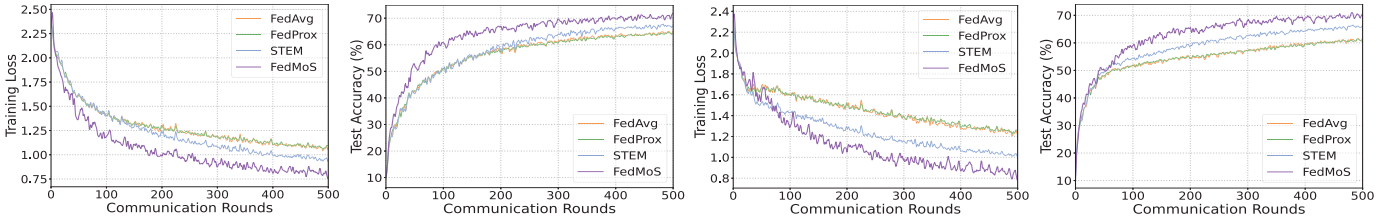
A. Evaluation Setup

1) Platform and Parameters: Evaluations are conducted on a Dell server with NVIDIA Tesla V100 GPUs using Pytorch.



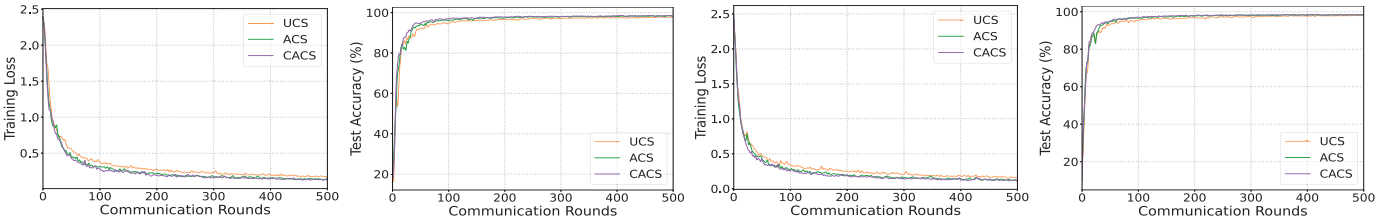
(a) Training loss over # round, $I = 5$ (b) Test accuracy over # round, $I = 5$ (c) Training loss over # round, $I = 10$ (d) Test accuracy over # round, $I = 10$

Fig. 2: Comparison results of FL algorithms on MNIST.



(a) Training loss over # round, $I = 5$ (b) Test accuracy over # round, $I = 5$ (c) Training loss over # round, $I = 10$ (d) Test accuracy over # round, $I = 10$

Fig. 3: Comparison results of FL algorithms on CIFAR-10.



(a) Training loss over # round, $I = 5$ (b) Test accuracy over # round, $I = 5$ (c) Training loss over # round, $I = 10$ (d) Test accuracy over # round, $I = 10$

Fig. 4: Comparison results of selection schemes for FedMoS on MNIST.

Specifically, we simulate $N = 500$ clients and a central PS, where out of 5% clients are selected in each round, i.e., $M = 25$. The sampled batch size is $B = 10$ and total communication round is $T = 500$ with extra rounds being $T_e = 4$. Also consider different settings of local steps, namely $I = 5$ and $I = 10$, to show the robustness of FedMoS.

2) Datasets and Models: We attain the results on two real-world datasets: MNIST [24] and CIFAR-10 [25]. In particular, MNIST contains 70000 handwritten images with each being a square $28 \times 28 = 784$ pixel gray-scale digit. CIFAR-10 consists of 60000 tiny ($32 \times 32 = 1024$ pixel) color images in 10 classes. Considering skewed non-i.i.d. data, we randomly subsample $D_i \in [10, 50]$ ($D_i \in [20, 200]$) training data from 2 (5) image classes on MNIST (CIFAR-10) for each client. Besides, we implement a convolutional neural network model followed by two fully connected layers for MNIST, and three fully connected layers for CIFAR-10.

3) Benchmark: To validate the proposed FedMoS, we introduce the following FL algorithms for comparison.

- FedAvg: federated averaging [5], which selects M out of N clients uniformly at random. Each selected client performs I-step SGD before being periodically synchronized by PS.
- FedProx: federated averaging with proximal term $\mu(\boldsymbol{x}_{i,\tau}^t - \boldsymbol{x}^t)$ in local update of uniformly sampled clients [6]. FedProx is developed to handle the data heterogeneity.
- STEM: stochastic two-sided momentum [19], which originally assumes a full client participation with uniform importance. To incorporate client sampling, we will first randomly select M clients and then follow STEM to train the model.

B. Performance of FedMoS

1) Comparison Results of FL Algorithms: First, we show that FedMoS with double momentum alone has superior performance over benchmarks. To this end, suppose all algorithms are based on *uniform client selection* (UCS) in Eq. (28).

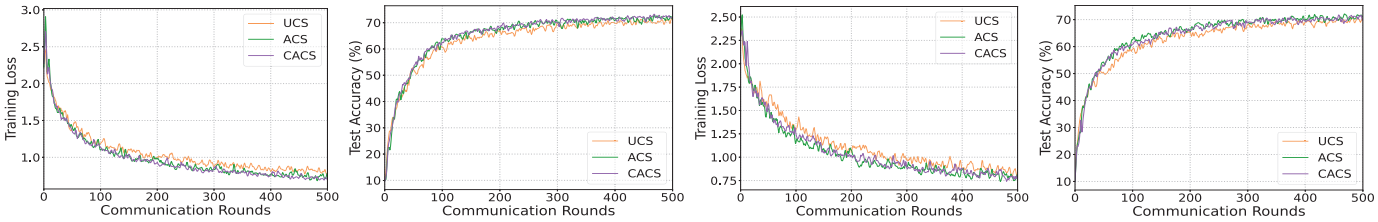
Training loss and test accuracy on MNIST are depicted in Fig. 2. Basically, the loss of FedMoS decreases much faster than FedAvg, FedProx, and STEM, i.e., FedMoS will converge to the stationary point more quickly. In the meantime, FedMoS can reach about 98% accuracy under $I = 5$ or $I = 10$, which is also the highest score achieved on MNIST. Similar observations are witnessed on CIFAR-10 in Fig. 3. Concretely, FedMoS yields an obviously *sharper* decline in global loss compared to FedAvg, FedProx, and STEM. Furthermore, we make an approximately 71.42% test accuracy using FedMoS on CIFAR-10, which *surpasses* the baselines by large margins, roughly 4%-9% higher under different local step settings.

2) Necessity of Adaptive Selection for FedMoS: We complete the picture by involving the sampling schemes, i.e., *adaptive client selection* (ACS) in Algorithm 2 and its variant, *clustered adaptive client selection* (CACS). As a comparison, we also exhibit the results of FedMoS with UCS.

Specifically, the performances regarding training loss and test accuracy on MNIST and CIFAR-10 are displayed in

TABLE I: Communication rounds for achieving target performance.

Dataset	Target	FedMoS (ACS)	FedMoS (CACS)	FedMoS (UCS)	FedAvg (UCS)	FedProx (UCS)	STEM (UCS)
MNIST	0.4 loss	62	54	80 (**1.5**×)	286 (3.6×, **5.3**×)	287 (3.6×, **5.3**×)	244 (3.1×, **4.5**×)
	95% acc.	62	51	90 (**1.8**×)	380 (4.2×, **7.5**×)	351 (3.9×, **6.9**×)	255 (2.8×, **5.0**×)
CIFAR-10	1.2 loss	76	73	84 (**1.2**×)	251 (3.0×, **3.4**×)	256 (3.0×, **3.5**×)	176 (2.1×, **2.4**×)
	60% acc.	76	71	88 (**1.2**×)	230 (2.6×, **3.2**×)	236 (2.7×, **3.3**×)	202 (2.3×, **2.8**×)



(a) Training loss over # round, $I = 5$ (b) Test accuracy over # round, $I = 5$ (c) Training loss over # round, $I = 10$ (d) Test accuracy over # round, $I = 10$

Fig. 5: Comparison results of selection schemes for FedMoS on CIFAR-10.

Figs. 4 and 5, respectively. In general, ACS and CACS have faster convergence rate pertaining to loss declines than UCS on both MNIST and CIFAR-10. As for the test accuracy, ACS and CACS achieve similar performance, approximately 98.61% on MNIST and 72.97% on CIFAR-10 under both local steps $I = 5$ and $I = 10$, which are higher than those of UCS.

C. Communications Required for Target Loss and Accuracy

We illustrate the communication rounds required for reaching the target performance, i.e., 0.4 (1.2) training loss and 95% (60%) test accuracy on MINIST (CIFAR-10). W.l.o.g., we mainly demonstrate the case where $I = 5$.

Results in Table I include benchmark FL algorithms (FedAvg, FedProx, STEM) with UCS, and FedMoS with UCS, ACS, and CACS. Particularly, we can see that FedMoS with UCS only (light numbers in brackets) mitigates the communications by 67%-72% (52%-67%) pertaining to the target loss, and by 65%-76% (56%-63%) regarding the target accuracy on MINIST (CIFAR-10). By involving adaptive client sampling (bold numbers in brackets), ACS/CACS expedites the convergence rate by 20%-80% than UCS for FedMoS, which is also a great improvement and seems counterintuitive to the relatively small distinctions in Figs. 4-5. Moreover, FedMoS with ACS/CACS is approximately *6 (3) times faster* than baselines on MNIST (CIFAR-10) in terms of the target loss or accuracy. Therefore, FedMoS is highly efficient with a remarkable reduction in communication overheads.

VI. Related Work

Momentum-based FL algorithms. Recently, FL has attracted a surge of attention where one of its distinct features is skewed non-i.i.d. data. Momentum is a promising technique to reduce the variance incurred by data heterogeneity [26]. MIME maintains the momentum on the client side as a combination of local variable and server state, so that client update can mimic a centralized training process [27]. In contrast, different forms of momentum on the server side are employed in [28] to increase the framework adaptivity. Incorporating the benefits of local and global momentum, Khanduri *et al.* propose STEM to minimize the training time and communication cost [19]. Nonetheless, existing double momentum-based FL algorithms are mainly built on uniform aggregation weight or full client participant, which are not general in practical scenarios.

Client selection. Perpendicular to momentum-based update, client selection is also widely adopted to reduce training time. Nishio *et al.* propose FedCS to pick clients according to their resource conditions, which needs many extra server-client interactions for exchanging client information [29]. Importance samplings with and without replacement are explored in [30], where both data and model variability would influence the convergence rate. Considering the existence of selection bias, Cho *et al.* design power-of-two strategies to achieve communication and computation efficient samplings [31]. Furthermore, Li *et al.* develop a sample-level selection to choose participants with high-quality data so as to improve learning performance [32]. However, existing sampling schemes are mostly based on basic FL algorithms, like FedAvg, while how to jointly design client selection and momentum buffer still remains open.

Communication-efficient methods. Communication efficiency can be empirically enhanced by precluding irrelevant updates in FL, since parameters often stabilize before the ultimate model convergence, i.e., reducing unnecessary synchronizations is effective to mitigate communication overheads [33], [34]. Besides, excluding stragglers with low system efficiency is also leveraged to promote the wall-time performance [35]. Though principled, they mainly have empirical improvements while lacking of theoretical guarantees.

VII. Conclusion

In this paper, we propose FedMoS, a *joint communication-efficient framework* of momentum-based update and adaptive client selection, to tame the client drift issue. We first maintain double momentum on both server and client sides to deal with the data heterogeneity. Through a rigid analysis, we characterize a *tight* $\mathcal{O}(T^{-2/3})$ convergence rate for FedMoS as long as client sampling is unbiased. Then, we design an adaptive client selection to optimally distribute the sampling probabilities. We show that the adaptive scheme can reduce the *sampling variance* while also maintaining *unbiased* model aggregation. Extensive evaluations conducted on real-world datasets corroborate the superiority of FedMoS over existing benchmarks in advocating FL convergence and accuracy.

REFERENCES

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM TIST*, vol. 10, no. 2, pp. 1-19, 2019.

[2] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated Learning for Mobile Keyboard Prediction," [Online]. Available: https://arxiv.org/abs/1811.03604

[3] T. H. Hsu, H. Qi, and M. Brown, "Federated Visual Classification with Real-World Data Distribution," in *Proc. ECCV*, 2020, pp. 76-92.

[4] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, "ClusterFL: A Similarity-Aware Federated Learning System for Human Activity Recognition," in *Proc. ACM MobiSys*, 2021, pp. 54-66.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. PMLR*, 2017, pp. 1273-1282.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," in *Proc. MLSys*, 2020, pp. 429-450.

[7] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating Federated Learning via Momentum Gradient Descent," *IEEE TPDS*, vol. 31, no. 8, pp. 1754-1766, 2020.

[8] H. Yuan and T. Ma, "Federated Accelerated Stochastic Gradient Descent," in *Proc. NeurIPS*, 2020, pp. 5332-5344.

[9] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning," in *Proc. ICML*, 2020, pp. 5132-5143.

[10] Y. Fraboni, R. Vidal, L. Kameni, and M. Lorenzi, "Clustered Sampling: Low-Variance and Improved Representativity for Clients Selection in Federated Learning," in *Proc. ICML*, 2021, pp. 407-3416.

[11] C. Briggs, Z. Fan, and P. Andras, "Federated Learning with Hierarchical Clustering of Local Updates to Improve Training on Non-IID Data," in *Proc. IEEE IJCNN*, 2020, pp. 1-9.

[12] Y. Fraboni, R. Vidal, L. Kameni, and M. Lorenzi, "On The Impact of Client Sampling on Federated Learning Convergence," [Online]. Available: https://arxiv.org/abs/2107.12211

[13] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling System and Statistical Heterogeneity for Federated Learning with Adaptive Client Sampling", in *Proc. IEEE INFOCOM*, 2022.

[14] F. Sattler, K. Müller, and W. Samek, "Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints," *IEEE TNNALS*, vol. 32, no. 8, pp. 3710-3722, 2021.

[15] R. Johnson and T. Zhang, "Accelerating Stochastic Gradient Descent using Predictive Variance Reduction," in *Proc. NeurIPS*, 2013.

[16] H. Yu, R. Jin, and S. Yang, "On the Linear Speedup Analysis of Communication Efficient Momentum SGD for Distributed Non-Convex Optimization," in *Proc. ICML*, 2019, pp. 97:7184-7193.

[17] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization," in *Proc. NeurIPS*, 2020.

[18] R. Das, A. Acharya, A. Hashemi, S. Sanghavi, I. S. Dhillon, and U. Topcu, "Faster Non-Convex Federated Learning via Global and Local Momentum," in *Proc. UAI*, 2022.

[19] P. Khanduri, P. Sharma, H. Yang, M. Hong, J. Liu, K. Rajawat, and P. K. Varshney, "STEM: A Stochastic Two-Sided Momentum Algorithm Achieving Near-Optimal Sample and Communication Complexities for Federated Learning," in *Proc. NeurIPS*, 2021, pp. 6050-6061.

[20] Z. Zhong, Y. Zhou, D. Wu, X. Chen, M. Chen, C. Li, and Q. Z. Sheng, "P-FedAvg: Parallelizing Federated Learning with Theoretical Guarantees," in *Proc. IEEE INFOCOM*, 2021, pp. 1-10.

[21] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the Convergence of FedAvg on Non-IID Data," in *Proc. ICLR*, 2020.

[22] Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. Woodworth, "Lower Bounds for Non-Convex Stochastic Optimization," *Mathematical Programming*, pp. 1-50, 2022.

[23] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing Federated Learning on Non-IID Data with Reinforcement Learning," in *Proc. IEEE INFOCOM*, 2020, pp. 1698-1707.

[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[25] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," [Online]. Available:http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf

[26] R. Kidambi, P. Netrapalli, P. Jain, and S. Kakade, "On the Insufficiency of Existing Momentum Schemes for Stochastic Optimization," in *Proc. IEEE ITA*, 2018, pp. 1-9.

[27] S. P. Karimireddy, M. Jaggi, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Mime: Mimicking Centralized Stochastic Algorithms in Federated Learning," [Online]. Available: https://arxiv.org/abs/2008.03606

[28] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konecny, S. Kumar, and H. B. McMahan, "Adaptive Federated Optimization," in *Proc. ICLR*, 2021.

[29] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," in *Proc. IEEE ICC*, 2019, pp. 1-7.

[30] E. Rizk, S. Vlaski, and A. H. Sayed, "Optimal Importance Sampling for Federated Learning," in *IEEE ICASSP*, 2021.

[31] Y. J. Cho, J. Wang, and G. Joshi, "Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies," in *Proc. AISTATS*, 2022.

[32] A. Li, L. Zhang, J. Tan, Y. Qin, J. Wang, and X. Li, "Sample-Level Data Selection for Federated Learning," in *Proc. IEEE INFOCOM*, 2021, pp. 1-10.

[33] L. Wang, W. Wang, and B. Li, "CMFL: Mitigating Communication Overhead for Federated Learning," in *Proc. IEEE ICDCS*, 2019, pp. 954-964.

[34] C. Chen, H. Xu, W. Wang, B. Li, B. Li, L. Chen, and G. Zhang, "Communication-Efficient Federated Learning with Adaptive Parameter Freezing," in *Proc. IEEE ICDCS*, 2021, pp. 1-11.

[35] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient Federated Learning via Guided Participant Selection," in *Proc. USENIX OSDI*, 2021, pp. 19-35.

[36] X. Wang, Y. Chen, Y. Li, X. Liao, H. Jin, and B. Li, "FedMoS: Taming Client Drift in Federated Learning with Double Momentum and Adaptive Selection," [Online]. https://wangxionghome.github.io/MainFL-TR.pdf