Fast Heterogeneous Federated Learning with Hybrid Client Selection

Abstract

Client selection schemes are widely adopted to handle communication-efficient problems in recent studies of Federated Learning (FL). However, the large variance of the model updates aggregated from the randomly-selected unrepresentative subsets directly slows the FL convergence. We present a novel clustering-based client selection scheme to accelerate the FL convergence by variance reduction. Simple yet effective schemes are designed to improve the clustering effect and control the effect fluctuation, therefore, generating the client subset with certain representativeness of sampling. Theoretically, we demonstrate the improvement of the proposed scheme in variance reduction. We also present the tighter convergence guarantee of the proposed method thanks to the variance reduction. Experimental results confirm the exceed efficiency of our scheme compared to alternatives.

1 INTRODUCTION

Federated Learning (FL) is a distributed learning paradigm for training a global model from data scattered across different clients Konečnỳ et al. [2015]. During the training process, all the clients need to operate data locally and transfer the model updates between servers and themselves back and forth. Such a training process may raise many challenges, with communication cost often being the critical bottleneck Kairouz et al. [2021].

Many studies have found that different clients might transfer similar (or redundant) model updates to the server, which is a waste of communication costs Balakrishnan et al. [2022], Fraboni et al. [2021], Karimireddy et al. [2020b]. Client selection schemes are widely adopted to reduce the waste of communication costs, e.g., Federated Averaging (`FedAvg`) where only the randomly-selected subset of clients transfer their model update to the server instead of yielding all clients involved McMahan et al. [2017]. `FedAvg` can maintain the learning efficiency with reduced communication costs, since the random selection schemes can reduce the redundant update transmission. When `FedAvg` meets client sets with low similarity, the server can not accurately pick out the representative clients. The large variance of the update gradient aggregated from the unrepresentative subset directly slows the training convergence Fraboni et al. [2021].

To accelerate the training convergence by variance reduction, many client selection criteria have been proposed in recent literature, e.g., importance sampling, where the probabilities for clients to be selected is proportional to their importance measured by the norm of update Chen et al. [2020], data variability Rizk et al. [2021], and test accuracy Mohammed et al. [2020]. However, importance sampling could not effectively capture the similarities among the clients. As shown in Figure 1(a), applying importance sampling could cause learning inefficiency as the clients transfer excessive important yet similar updates to the server. To make better use of the similarities among clients, some researchers propose raw gradient-based cluster sampling schemes. As shown in Figure 1(b), they first group the clients with similar model updates together Fraboni et al. [2021], Muhammad et al. [2020], and apply random selection within each cluster. However, cluster sampling schemes only work under the strong assumption that we can have a decent clustering effect all the time, i.e., almost all clusters have small intra-cluster distances, with little variation from cluster to cluster. Unfortunately, the effect of raw gradient-based clustering methods faces the following problems: **(i) Poor Effect**, the high dimension gradient of a client is too complicated to be an appropriate cluster feature, which can not bring small intra-cluster distances. **(ii) Effect Fluctuation**, due to the limitations of the clustering algorithm, the clustering effect tends to fluctuate greatly, i.e., clusters tend to have diverse heterogeneity. In practice, such diverse heterogeneity always leads to the intra-cluster distances of different clusters varying greatly. After clustering with poor and fluctuant effects,

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

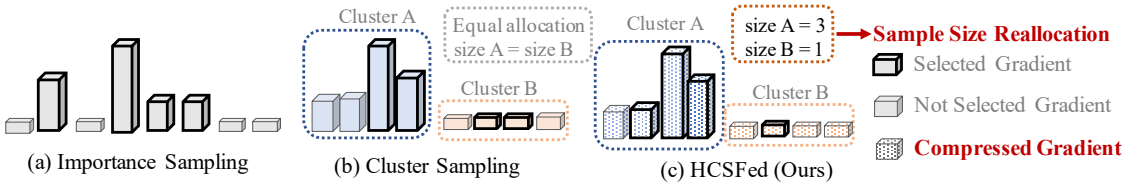


Figure 1: The differences among client selection schemes. The heights represent the norm of the gradients. The different colors denote different clusters. We highlight the main difference by red bold font.

there always exists some clusters with low client similarity (please refer to Figure. 2). Applying client selection in the clusters with low client similarity may slow the overall training convergence. Besides, cluster sampling methods require all the clients to return the raw gradient for better selection, which runs against the communication reduction objective.

We propose **`HCSFed`**, a novel **H**ybrid **C**lustering-based client **S**election scheme for heterogeneous **Fed**erated learning that further accelerates the FL convergence. To improve the **poor clustering effect**, different from using the high dimension gradient, we adopt the compressed gradient as the cluster feature. Adopting the compressed gradient as a cluster feature not only allows a better clustering effect but also is communication-efficient since the compression operation filters out the redundant information within each gradient. Besides, to control the **effect fluctuation**, we reset the number of clients to be selected in each cluster based on the client similarity and the size of the cluster, which we term as sample size re-allocation. As shown in Figure 1(c), the sample size re-allocation scheme makes the cluster with low similarity have more clients to be selected. The sampling effect becomes stable because the poor effect of random sampling in the clusters with low similarity is greatly mitigated by the sample size re-allocation scheme. Theoretically, we demonstrate the improvement `HCSFed` in variance reduction. Extensive experiments show the exceed efficiency of the proposed scheme compared to the cutting-edge FL client selection criteria in both convex and non-convex settings. Our main contributions are summarized below:

- To the best of our knowledge, this work provides the first insight into modeling the diverse heterogeneity of clusters which leads to the client sampling effect fluctuation. (please refer to our sample size re-allocation part section 3)
- We present the explicit cross-client variance comparison among different client selection methods which is a new perspective to watch the convergence improvement of the advanced selection method. (please refer to our theoretical analysis part section 4)
- A series of experiments in different settings verify the effectiveness of our proposed method and the correctness of our theoretical results.

2 SYSTEM SETUP AND PRIOR WORK

The Federated Learning Optimization Settings. In federated learning, a total number of N clients aim to jointly solve the following distributed optimization model:

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) := \sum_{k=1}^{N} \omega_k F_k(\mathbf{w}) := \frac{1}{n_k}\sum_{j=1}^{n_k} f(\mathbf{w}; x_{k,j}), \quad (1)$$

where ω_k is the weight of the kth client, $s.t.\ \omega_k \geq 0$ and $\sum_{k=1}^{N}\omega_k = 1$. Suppose the kth client possesses n_k training data: $\{x_{k,1},\ x_{k,2},\ \ldots,\ x_{k,n_k}\}$. The local objective function $F_k(\mathbf{w})$ is defined as the average of $f(\mathbf{w}; x_{k,j})$, which is a user-specified loss function (possibly non-convex) made with model parameters $\mathbf{w} \in \mathbb{R}^d$ and training data $x_{k,j}$.

`FedAvg` Description. In the tth train round of the `FedAvg`, the server broadcasts the latest global model $\mathbf{w}_t$ to all the clients. The kth client sets $\mathbf{w}_t^k \leftarrow \mathbf{w}_t$ and performs local training for E epochs:

$$\mathbf{w}_{t+1}^k \longleftarrow \mathbf{w}_t^k - \eta_t \nabla F_k\left(\mathbf{w}_t^k, \xi_t^k\right), \quad (2)$$

where η_t is the learning rate and ξ_t^k is a data sample uniformly selected from the local data. In the communication round, the server aggregates the local model update to produce the new global model $\mathbf{w}_{t+E}$. In fact, to reduce the communication cost, `FedAvg` randomly generates a subset $\mathcal{S}_t$ consisting of m clients from the entire client set and aggregates the model update, $\{\mathbf{w}_{t+E}^1, \mathbf{w}_{t+E}^2, \cdots, \mathbf{w}_{t+E}^m\}$,

$$\mathbf{w}_{t+E} \longleftarrow \underbrace{\sum_{k=1}^{N}\omega_k \mathbf{w}_{t+E}^k}_{\text{Full Participate}} \longleftarrow \underbrace{\sum_{k\in\mathcal{S}_t}\frac{N}{m}\omega_k \mathbf{w}_{t+E}^k}_{\text{Partial Participate}}. \quad (3)$$

`LAG` Chen et al. [2018] is a improved aggregation scheme where it triggers the reuse of outdated gradients for the non-selected clients, but the variance of the aggregated gradient is still large, which directly slows the FL convergence. To accelerate the convergence by variance reduction, many client selection criteria have been proposed in recent literature, including importance-based and clustering-based methods.

Importance Sampling. Importance sampling is a non-uniform client selection method, where the probabilities

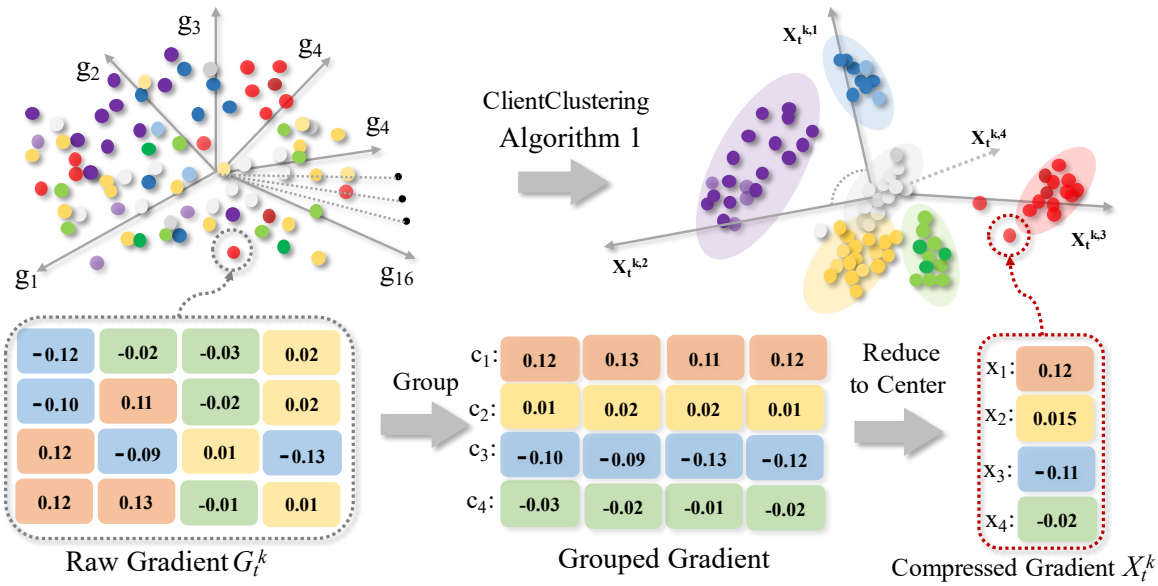


Figure 2: The visualizations of the raw gradient (left) and the clustered compressed gradient (right). Each point denotes the update gradient of one client. The g_i is the ith component of the update gradient. The c_i ci denotes the ith group of component.

for clients to be chosen are proportional to their importance Katharopoulos and Fleuret [2018], Zhao and Zhang [2015]. Different methods to measure the importance are proposed, including data variability Rizk et al. [2021], norm of update Chen et al. [2020], test accuracy Mohammed et al. [2020], local rounds Singh et al. [2019], and local loss Cho et al. [2020]. Such methods can yield better selection by activating the clients with more importance. However, they could not capture the similarities among the updates of clients, therefore, triggering the communication redundancy.

Cluster Sampling. To make better use of similarities among clients, some researchers propose raw gradient-based cluster sampling schemes that group the clients with similar gradients together Fraboni et al. [2021], Muhammad et al. [2020]. However, the effect of cluster sampling strongly relies on the performance of the clustering process. Unfortunately, the high dimension gradient of a client is too complicated to be an appropriate cluster feature, which can not bring a good clustering effect. Besides, meticulous one-by-one client selection Balakrishnan et al. [2022], Dieuleveut et al. [2021] is introduced to FL. It could achieve better selection outcomes but with extra expensive computation for solving each noncommittal submodular maximization. We want to claim that cluster sampling needs to transmit the raw gradient information of all the clients to the server, which would bring unacceptable communication costs. And if the server has received all the updates, why don't we aggregate them immediately? When all the clients participate in the training, there is no cross-client anymore.

Difference from `SCAFFOLD`. While `SCAFFOLD` Karimireddy et al. [2020b], Gorbunov et al. [2021], Karimireddy et al. [2020a] appears to be similar to our method, there are fundamental differences. `SCAFFOLD` augments the updates with extra "control-variate" item that is also transmitted along with the updates to reduce the variance, while we reduce the variance by selecting a more representative subset. The former focuses on update control, while the latter focuses on better update selection, indicating that they are orthogonal and compatible with each other.

3 METHOD

How to explore and make use of the similarities among clients with acceptable resource costs is the key issue in constructing client selection criteria. We propose `HCSFed`, a selection scheme that exploits three components (cluster sampling, sample size re-allocation, importance sampling) guaranteeing convergence speedup by variance reduction.

The Compressed Gradient-Based Cluster Selection. To make use of the similarity among clients, we develop cluster selection that groups the similar clients together based on the client characteristic. The following is a mathematical description of cluster selection. Generating a subset $\mathcal{S}_t$ consisting of m clients by clustering selection can be regarded as selecting m clients using H different probability distributions $\{P_h\}_{h=1}^{\mathrm{H}}$ where $P_h = \{p_k^h\}_{k=1}^{N}$. For example, if we select a client from the hth cluster, the probability

distribution P_h will be used, and the p_k^h comes as follows:

$$p_k^h = \begin{cases} 0, & \text{if } k^{th} \text{ client} \notin h^{th} \text{ cluster} \\ \frac{1}{N_h}, & \text{if } k^{th} \text{ client} \in h^{th} \text{ cluster} \end{cases} \tag{4}$$

where N_h denotes the number of clients in the hth cluster $s.t. \sum_{h=1}^{\mathrm{H}} N_h = N$. Prior cluster selections Fraboni et al. [2021] prefer to develop clustering based on the raw high dimension gradient (shown in Figure 2), which is too complicated to distinguish clients and brings unacceptable communication costs running against the original objective. Different from the previous raw gradient-based clustering methods, we develop clustering based on compressed gradient Han et al. [2015] which is the information-preserving representation of the model update. As shown in Figure 2, different components of the gradient are grouped based on their numerical value[1], only the center of the group are retained to form the compressed gradient. The compression rate R can be obtained by evaluating the quotient of d and d', i.e., $\frac{d'}{d}$, where d' and d are the total dimension of the compressed gradient and raw gradient, respectively. Adopting the compressed gradient as cluster feature not only allows better clustering effect but also is communication-efficient, since the compression operation filters out the redundant information within each gradient. The pseudo-code of Gradient Compress (GC) can be found in Appendix A.2. Algorithm 1 summarizes the main steps of client clustering. Ideally, after clustering, the clients in the same cluster are similar to each other. However, as shown in Figure 2, some clusters always exist with low similarity (purple cluster), since the effect of the clustering process fluctuates, which may also lead to performance degradation.

Algorithm 1: ClientClustering

Input: compressed gradient of all the clients in the tth round $\{X_t^k\}_{k=1}^N$
Input: The number of the clusters H
Initialize Randomly select H clients as cluster centers $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_{\mathrm{H}}\}$;
Initialize $C_i = \varnothing \ (1 \leq i \leq \mathrm{H})$;
repeat
  for *each client k=1,2,...,N* **do**
    $\lambda_k = \arg\min_{i \in \{1,2,\ldots,\mathrm{H}\}} \left\| X_t^k - \boldsymbol{\mu}_i \right\|_2$;
    $C_{\lambda_k} = C_{\lambda_k} \bigcup \{k^{th} \text{ client with } X_t^k\}$;
  end
  for *each cluster i=1,2,...,*H **do**
    New center $\boldsymbol{\mu}_i' = \frac{1}{|C_i|} \sum_{X_t^k \in C_i} X_t^k$;
    $\boldsymbol{\mu}_i \leftarrow \boldsymbol{\mu}_i'$;
  end
until $\forall\, i = \{1, 2, \ldots, \mathrm{H}\}, \boldsymbol{\mu}_i' = \boldsymbol{\mu}_i$;
Output: $\mathcal{G} = \{C_1, C_2, \ldots, C_{\mathrm{H}}\}$;

Algorithm 2: HCSFed

Input: updates in the tth round of all clients $\{G_t^k\}_{k=1}^N$
Initialize $\mathbf{w}_0$;
Initialize $\mathcal{S}_t = \varnothing$;
Server executes:
for *each round* $t = 1, 2, \cdots$ **do**
  $m \leftarrow \max(qN, 1)$;
  $\mathcal{G} \leftarrow$ ClientClustering $(\mathbf{X}_t, \mathrm{H})$;
  for *each cluster* $h = 1, 2, \cdots, \mathrm{H}$ **do**
    $m_h \leftarrow \frac{N_h S_h}{\sum_{h=1}^{\mathrm{H}} N_h S_h} \cdot m$;
    compute $\{p_t^k\}_{k \in N_h}$ using Equation 6;
    $\mathcal{S}_t = \mathcal{S}_t \bigcup m_h$ clients selected with P_h;
  end
  for *client* $\in \mathcal{S}_t$ ***in parallel*** **do**
    $\mathbf{w}_{t+1}^k \leftarrow$ ClientUpdate$(k, \mathbf{w}_t^k)$;
  end
  $\mathbf{w}_{t+1} \leftarrow \sum_{k \in \mathcal{S}_t} \frac{N}{m} \omega_k \mathbf{w}_{t+1}^k$;
end
ClientUpdate$(k, \mathbf{w}_t^k)$:
for *each local epoch i from* 1 *to* E **do**
  $\mathbf{w}_{t+i+1}^k \longleftarrow \mathbf{w}_{t+i}^k - \eta \nabla F_k \left(\mathbf{w}_{t+i}^k, \xi_{t+i}^k\right)$;
end
if $k \in \mathcal{S}_t$ **then**
  return $\mathbf{w}_{t+1}^k \leftarrow \mathbf{w}_{t+E}^k$, $X_t^k \leftarrow \mathrm{GC}(G_t^k)$;
else
  return $X_t^k \leftarrow \mathrm{GC}(G_t^k)$;
end

Sample Size Re-allocation Scheme. Concentrating on the diverse heterogeneity of clusters, we make the first attempt to develop a sample size re-allocation scheme that pays more attention to the cluster with great heterogeneity. HCSFed redetermines the sample size m_h, namely, the number of clients sampled from the hth cluster, by considering both the cluster size N_h and variability.

$$m_h = \frac{N_h S_h}{\sum_{h=1}^{\mathrm{H}} N_h S_h} \cdot m, \tag{5}$$

where $S_h = \frac{1}{N_h - 1} \sum_{j=1, i \neq j}^{N_h} \left\| X_t^i - X_t^j \right\|_2^2$, m_h denotes the number of clients in the subset $\mathcal{S}_t$ from the hth cluster $s.t. \sum_{h=1}^{\mathrm{H}} m_h = m$. S_h denotes the variability of the hth cluster where we use Cluster Cohesion based on the compressed model update to approximate. The introduction of sample size re-allocation can further reduce the variance by assigning more sample chance to those clusters with greater heterogeneity. Due to the limited total number of clients to be selected, sometimes some clusters with great heterogeneity still can not have an adequate sample chance, which indeed need extra care.

[1]Typical approaches like PCA cannot be adopted in our work: each client operates data locally, which means reduced dimension d' is strictly lower than the min value of the number of samples m and the number of raw dimension d.

Importance Selection. We introduce importance selection to optimize the probabilities for clients to be selected in one cluster, which is based on the norm of the compressed gradient, i.e., $\mathrm{GC}(G_t^k) = X_t^k$. We re-determine the probability for each client to be sampled in t^{th} round as follows,

$$p_t^k = \frac{\left\|\mathrm{GC}(G_t^k)\right\|}{\sum_{k=1}^{N_h}\left\|\mathrm{GC}(G_t^k)\right\|} = \frac{\left\|X_t^k\right\|}{\sum_{k=1}^{N_h}\left\|X_t^k\right\|}, \quad \text{if } k^{th} \text{ client} \in h^{th} \text{ cluster in } t^{th} \text{ round.} \tag{6}$$

After completing the importance selection, the client with a higher norm of gradient (more importance) will have a higher probability to be sampled, which brings representative outcomes under an inadequate sampling ratio. Importance selection provides a fine-grained optimization over cluster selection with sample size re-allocation, i.e., assigning more attention to the more representative clients.

These ideas of selection discussed in this subsection together constitute the `HCSFed` to optimize the client selection in FL. `HCSFed` selects representative clients via allocating appropriate probability for each client to be selected, with the promise of reducing the variance between the model update aggregated from the sampled subset $\mathcal{S}_t$ and the entire set $\mathcal{K}$. Algorithm 2 summarizes the main steps of `HCSFed`.

4 THEORETICAL ANALYSIS

4.1 VARIANCE RELATIONSHIP AMONG SELECTION SCHEMES

Theorem 1 (Variance Reduction)**.** *If the population is large compared to the subset, $\frac{m}{N}$, $\frac{m_h}{N_h}$, $\frac{1}{m_h}$ and $\frac{1}{N}$ are negligible, then the cross-client variance of different selection schemes satisfy*

$$\mathbb{V}\left(\mathbf{w}_{hybrid}\right) \leq \mathbb{V}\left(\mathbf{w}_{cludiv}\right) \leq \mathbb{V}\left(\mathbf{w}_{cluster}\right) \leq \mathbb{V}\left(\mathbf{w}_{rand}\right),$$

where $\mathbf{w}_{hybrid}, \mathbf{w}_{cludiv}, \mathbf{w}_{cluster}, \mathbf{w}_{rand}$ denote the model update aggregated from the subset $\mathcal{S}_t$ that generated by `HCSFed`, clustering selection scheme under re-allocation, clustering selection scheme under plain allocation and simple random selection scheme, respectively.

Proof Sketch. Below we provide a proof sketch to reveal the relationship among the variance of different client selection schemes, and then we naturally state when our `HCSFed` scheme could achieve variance reduction. We defer the details of proof to Appendix B.

Comparison of Random and Clustering Selection. We derive the Equation 7 to show the relationship between $\mathbb{V}\left(\mathbf{w}_{cluster}\right)$ and $\mathbb{V}\left(\mathbf{w}_{rand}\right)$:

$$\mathbb{V}\left(\mathbf{w}_{rand}\right) = \mathbb{V}\left(\mathbf{w}_{cluster}\right) + \sum_{h=1}^{\mathrm{H}} \frac{N_h \left\|\mathbf{W}_h - \mathbf{W}(\mathcal{K})\right\|_2^2}{mN}, \tag{7}$$

where $\mathcal{K}$ denotes the set of all the clients. We have $\mathbb{V}\left(\mathbf{w}_{cluster}\right) < \mathbb{V}\left(\mathbf{w}_{rand}\right)$, unless $\forall h \in \{1, \cdots, \mathrm{H}\}, \mathbf{W}_h = \mathbf{W}(\mathcal{K})$, i.e., each cluster has the same averaged model update with the entire population, which indicates that all the clusters are homogeneous in terms of the mean model update.

Comparison of Plain Clustering and Clustering with Re-allocation. We derive the Equation 8 to show the relationship between $\mathbb{V}\left(\mathbf{w}_{cludiv}\right)$ and $\mathbb{V}\left(\mathbf{w}_{cluster}\right)$:

$$\mathbb{V}\left(\mathbf{w}_{cluster}\right) = \mathbb{V}\left(\mathbf{w}_{cludiv}\right) + \sum_{h=1}^{\mathrm{H}} \frac{N_h (S_h - S)^2}{mN}. \tag{8}$$

We have $\mathbb{V}\left(\mathbf{w}_{cludiv}\right) < \mathbb{V}\left(\mathbf{w}_{cluster}\right)$, unless $\forall h \in \{1, \cdots, \mathrm{H}\}, S_h = S$, i.e., each cluster has equal variability, which indicates that all the clusters are homogeneous in terms of the variability.

Comparison of Clustering with Re-allocation and `HCSFed`. We derive the Equation 9 to show the relationship between $\mathbb{V}\left(\mathbf{w}_{cludiv}\right)$ and $\mathbb{V}\left(\mathbf{w}_{hybrid}\right)$:

$$\mathbb{V}\left(\mathbf{w}_{cludiv}\right) = \mathbb{V}\left(\mathbf{w}_{hybrid}\right) + \frac{\left(\sum_{i=1}^{N_h}\left\|G_i\right\|_2\right)^2}{N_h} \sum_{i=1}^{N_h}\left(I_i - \frac{1}{N_h}\right)^2. \tag{9}$$

We have $\mathbb{V}\left(\mathbf{w}_{hybrid}\right) < \mathbb{V}\left(\mathbf{w}_{cludiv}\right)$, unless $\forall i \in \{1, \cdots, N_h\}, I_i = \frac{\|G_i\|_2}{\sum_{i=1}^{N_h}\|G_i\|_2} = \frac{1}{N_h}$, i.e., each client in h^{th} cluster has equal norm of model update.

Remark 1. *Theorem 1 verifies the capability of `HCSFed` in variance reduction. The proof sketch indicates that the variance difference of different selections will vanish only if all the clusters and the norm of the gradients are identical. The proposed approach can achieve variance reduction, since each cluster never has identical update mean, variability, and norm in heterogeneous FL.*

4.2 CONVERGENCE BEHAVIOR OF `HCSFED`

We emphasize that the convergence analysis of `FedAvg` with random selection is **NOT** our contribution, we just follow the previous analysis work Li et al. [2019] to verify that `HCSFed` could maintain the same convergence guarantees as a random selection in the worst case and in most cases, `HCSFed` will enjoy tighter convergence.

We next state the assumptions used in our theorem and proof, which are common in FL optimization literature, e.g., Zhou and Cong [2017], Li et al. [2019], Wang and Joshi [2021], Stich [2018], Haddadpour and Mahdavi [2019], Haddadpour et al. [2019]. Assume each function $F_k (k \in [N])$ is μ-strongly convex and L-smooth. Suppose that for all $k \in [N]$

Table 1: Required rounds for different methods with convex model (logistic regression) to achieve 80% accuracy on non-IID MNIST and FMNIST. 200+ indicates 80 % accuracy was not reached after 200 rounds.

	Methods	Sampling Ratio 10%		Sampling Ratio 30%		Sampling Ratio 50%	
		Num. of Rounds	Speedup	Num. of Rounds	Speedup	Num. of Rounds	Speedup
MNIST	Random	96	(1.0×)	42	(1.0×)	28	(1.0×)
	SCAFFOLD	74	(1.3×)	36	(1.2×)	21	(1.3×)
	Importance	35	(2.7×)	24	(1.8×)	22	(1.3×)
	Cluster	31	(3.1×)	16	(2.6×)	8	(3.5×)
	HCSFed	**9**	(**10.7**×)	**8**	(**5.2**×)	**8**	(**3.5**×)
FMNIST	Random	200+	(1.0×)	200+	(1.0×)	158	(1.0×)
	SCAFFOLD	180	(>1.1×)	144	(>1.4×)	115	(1.4×)
	Importance	180	(>1.1×)	116	(>1.7×)	100	(1.6×)
	Cluster	145	(>1.4×)	104	(>1.9×)	75	(2.1×)
	HCSFed	**81**	(>**2.5**×)	**80**	(>**2.5**×)	**75**	(**2.1**×)

and all t, the variance and expectation of stochastic gradients in each client on random samples ξ are bounded by σ_k^2 and G^2, i.e., $\mathbb{E}\left[\left\|\nabla F_k\left(\mathbf{w}_t^k, \xi\right) - \nabla F_k\left(\mathbf{w}_t^k\right)\right\|_2^2\right] \leq \sigma_k^2$ and $\mathbb{E}\left[\left\|\nabla F_k^2\left(\mathbf{w}_t^k, \xi\right)\right\|_2^2\right] \leq G^2$, respectively. And Γ is used to quantify the heterogeneity, where $\Gamma = F^* - \sum_{k=1}^N p_k F_k^*$.

Theorem 2 (Convergence Bound)**.** *Let Assumptions above hold and L, μ, σ_k, G be defined therein. Consider* `FedAvg` *when sampling m clients, then* `HCSFed` *satisfies:*

$$\mathbb{E}\left[F\left(\mathbf{w}_T\right)\right] - F^* \leq \underbrace{\mathcal{O}\left(\frac{\sum_k p_k^2 \sigma_k^2 + (E^2+\gamma)G^2 + L\Gamma}{\mu T}\right)}_{Bounding\ Full\ Participation} + \underbrace{\mathcal{O}\left(\frac{E^2 G^2}{m\mu T}\right)}_{Variance} \tag{10}$$

Remark 2. *Theorem 2 Li et al. [2019] gives the convergence upper bound of* `FedAvg` *with hybrid client selection. The first term is used to bound the* `FedAvg` *with full client participant, while the second term is used to bound the variance of the model update aggregated from the subset. In Theorem 1, we demonstrate the effectiveness of* `HCSFed` *in variance reduction.* `HCSFed` *achieves a lower variance, which indicates that* `HCSFed` *enjoy a tighter convergence bound.*

Remark 3. *Theorem 2 Li et al. [2019] uses Lemma 1-3 to bound the error of* `FedAvg` *with full participation corresponding to the first term, while using Lemma 4 to verify the unbiased property and Lemma 5 to bound the variance resulting from the client selection. Naturally,* `HCSFed` *satisfies the Lemma 5, since the Theorem 1 shows that* `HCSFed` *could achieve variance reduction. The dependency between the Lemmas and Theorem 2 indicates that the proof the Lemma 4 is enough to maintain the convergence bound. We defer the proof to Appendix A.3.*

We demonstrate the convergence guarantee of `HCSFed` under μ-strongly convex assumption. As for the non-convex loss function, we refer to the previous proved proposition that `FedAvg` with any unbiased selections maintains the same FL convergence bound of random selection (Theorem 2 in Fraboni et al. [2021]).

5 EXPERIMENT

5.1 EXPERIMENTAL SETUP

We run logistic regression (convex) and a fully connected network with one hidden layer of 50 nodes (non-convex) on MNIST [LeCun, 1998]. As for CIFAR-10 [Krizhevsky et al., 2009] and FMNIST [Xiao et al., 2017], we use the same classifier of `FedAvg` [McMahan et al., 2017] composed of 3 convolutional and 2 fully connected layers. We partition the dataset into 100 clients under both IID and non-IID data distribution. IID data partition strategy employs the idea of random split to create a uniform federated dataset. As for non-IID data partition, we use Dirichlet distribution to partition the entire client set. Dirichlet distribution, i.e., $\mathrm{Dir}(\alpha)$, gives to each client the respective partitioning across labels by changing the value of α. Specifically, the lower the value of α, the more heterogeneous the dataset is. The schematic table of the Dirichlet distribution is presented in Appendix A.1. As for the hyperparameters, N is the number of all clients, q is the sampling ratio, $nSGD$ is the times of SGD running locally, η is the learning rate, B is the batch size, T is the terminate round. For better comparison, we set all the hyperparameters following the FL optimization work `FedProx` [Li et al., 2020].

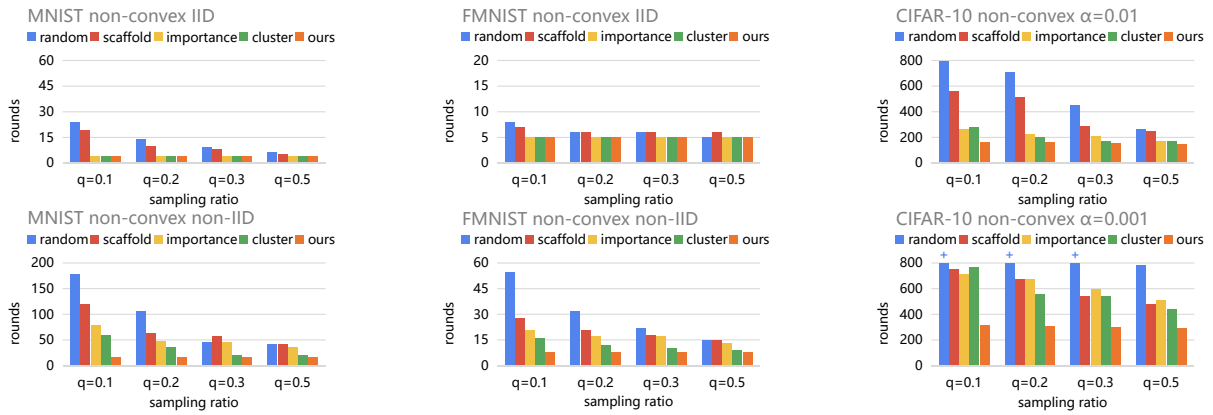


Figure 3: Required rounds for random, cluster, importance sampling, SCAFFOLD, and HCSFed to achieve 60% accuracy with $q = 0.1$, $N = 100$, $nSGD = 50$, $\eta = 0.01$, $B = 50$ on MNIST and FMNIST, with $\alpha \in \{0.01, 0.001\}$, $q = 0.1$, $N = 100$, $nSGD = 80$, $\eta = 0.05$, $B = 50$ on CIFAR-10.

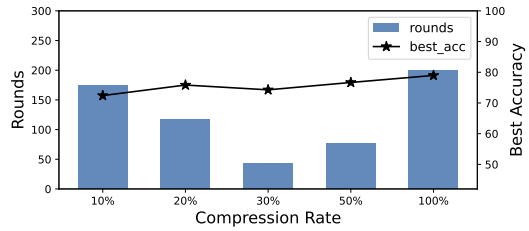
5.2 MAIN RESULTS

Our evaluation target is to compare the convergence speed of different methods in convex and non-convex FL context. We choose image classification as our downstream task and compare HCSFed with simple random sampling [McMahan et al., 2017], norm-based importance sampling [Chen et al., 2020], cluster sampling [Fraboni et al., 2021], and SCAFFOLD [Karimireddy et al., 2020b]. For better representation of the convergence speed, we follow the previous FL acceleration work setting [Karimireddy et al., 2020b] and report the required rounds for different methods with convex model to reach the target test accuracy in Table 1. We can observe that HCSFed consistently achieves the fastest convergence (has the fewest required rounds to reach the target accuracy) across all settings, surpassing not only the client selection schemes but also the SOTA variance-reduced method SCAFFOLD [Karimireddy et al., 2020b].

For better representation of the convergence speed, we follow the previous FL acceleration work setting [Karimireddy et al., 2020b] and report the required rounds for different methods with convex model to reach the target test accuracy in Table 1. We can observe that HCSFed consistently achieves the fastest convergence (has the fewest required rounds to reach the target accuracy) across all settings, surpassing not only the client selection schemes but also the SOTA variance-reduced method SCAFFOLD [Karimireddy et al., 2020b]. We also run HCSFed with non-convex model on different datasets. The result of non-convex experiment is presented in Figure 3. HCSFed consistently requires the fewest rounds to reach the target accuracy. All the results reveal that HCSFed can greatly accelerate the convergence, especially in the low sampling ratio and date heterogeneity case. As we emphasized in the theoretical analysis, the improvement of HCSFed depends on the sampling ratio and the client heterogeneity.

The lower the sampling ratio and the more heterogeneous the dataset, the more improvement HCSFed can enjoy. Naturally, such great improvement in convergence speed can be attributed to the two key components in HCSFed: (i) Different from the previous raw gradient-based methods, we introduce gradient compression to improve the clustering effect by using the more expressive cluster feature representation. (ii) To further guarantee the robustness of the sampling effect, we introduce a sample size re-allocation scheme and extra importance sampling to control the sampling effect fluctuation in the cluster with low client similarity. For more details, please refer to Appendix C.

(a) The impact of different numbers of clusters



(b) The impact of different compression rates

Figure 4: Visualization of Sensitivity Study Results.

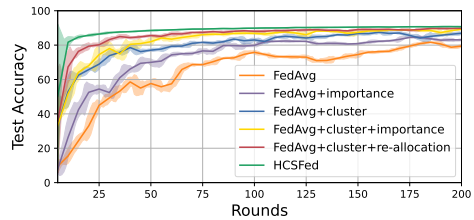
5.3 PARAMETER SENSITIVITY STUDY

The Number of Clusters (H). This experiment demonstrates the sensitivity of `HCSFed` to the different numbers of clusters. We run convex model-based `HCSFed` on non-IID MNIST with different numbers of clusters, i.e. with $\mathrm{H} \in \{5, 6, \cdots, 10\}$. We report the best classification accuracy and the required rounds for `HCSFed` with different numbers of clusters to reach the best test accuracy in Figure 4(a). `HCSFed` achieves stable classification accuracy and similar convergence speeds (measured by the required rounds to achieve the best accuracy), which highlights the robust efficiency of `HCSFed`. The results also reveal that the clustering effect of `HCSFed` is robust regardless of the number of clusters. Such merit might attribute to the sample size re-allocation scheme or the importance sampling scheme that we use to control the clustering effect fluctuation.

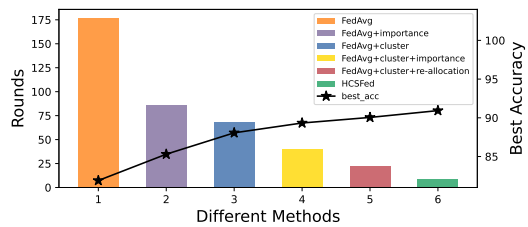
The Compression Rate (R). We also evaluate how different compression rates affect the global model performance. We run convex model-based `HCSFed` on non-IID MNIST with different compression rates (R), i.e. with $\mathrm{R} \in \{5\%, 10\%, 20\%, 40\%, 100\%\}$. We report the best accuracy and the required rounds of `HCSFed` with different compression rates to reach the best test accuracy in Figure 4(b). The results show that `HCSFed` achieves different convergence speed as the compression rate changes. The noticeable information is that both too high and too low compression rates can not achieve good learning efficiency. Intuitively, the low compression rate can not ensure the integrity of gradient information while the high compression rate is too complicated to learn efficiently. When we set the compressed rate to 100%, i.e., without gradient compression, the results show that both the convergence speed and final accuracy of the model are not satisfactory, which confirms the effectiveness of gradient compression. In practice, following the previous compression work Han et al. [2015], we set the compression rate with the highest within-group sum of squares to get a good sampling effect.

5.4 ABLATION STUDY

We run logistic regression (convex) on non-IID MNIST with $q = 0.1, nSGD = 50$ to analyze the individual contributions of different components of `HCSFed` in both final accuracy and convergence speed. In Figure 5(a), `HCSFed` is represented as the green line starting from the accuracy of 10% and ending at 91%, whereas `FedAvg` is the yellow line starting also from the same position as `HCSFed` but ending at 81% after 200 rounds, i.e., 10% less. We report the required rounds for different methods to achieve 80% accuracy in Figure 5(b). Compared with the plain `FedAvg`, `FedAvg` combined with cluster (blue) or importance sampling (purple) can accelerate the convergence speed (measured by the required rounds to achieve the target accuracy) by $2.6\times$ and $2.1\times$ times. The experimental results confirm the important role of the two in improving learning efficiency. Impressively, compared with importance sampling, the improvement of the clustering process is more obvious. This is because clustering can make better use of the correlation among clients, thereby selecting more representative clients to participate in model training. We also compare `FedAvg` combined with cluster, `FedAvg` combined with cluster and re-allocation, `FedAvg` combined with cluster and importance sampling, and `HCSFed`. All the latter methods can achieve a faster convergence in model training than the former, which validates the effectiveness of re-allocation and extra importance sampling. These results indicate that each component of `HCSFed` is complementary to each other, and also show the correctness of the variance reduction theory.



(a) The test accuracy of individual components



(b) The impact of individual components

Figure 5: Visualization of Ablation Study Results.

6 CONCLUDING REMARKS

In this paper, we propose `HCSFed`, a novel clustering-based selection scheme to accelerate the training convergence by variance reduction. Theoretically, we demonstrate the improvement of the proposed scheme in variance reduction. We present the convergence guarantee of `HCSFed` under the convex assumption. Experimental results demonstrate the superiority and the effectiveness of our method with both the convex and non-convex models. In the future, the co-variance among different clients and unavailable client settings will be considered, since it may also lead to performance degradation in FL. In the theoretical aspect, following the latest convergence analysis Khaled et al. [2020], we plan to loose the widely used but strict assumption "bounded gradient" and convexity for more general analysis of the faster convergence achieved by `HCSFed`.

References

Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. Diverse client selection for federated learning via submodular maximization. In *International Conference on Learning Representations*, 2022.

Tianyi Chen, Georgios Giannakis, Tao Sun, and Wotao Yin. Lag: Lazily aggregated gradient for communication-efficient distributed learning. *Advances in Neural Information Processing Systems*, 2018.

Wenlin Chen, Samuel Horvath, and Peter Richtarik. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723*, 2020.

Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020.

Aymeric Dieuleveut, Gersende Fort, Eric Moulines, and Geneviève Robin. Federated expectation maximization with heterogeneity mitigation and variance reduction. *arXiv preprint arXiv:2111.02083*, 2021.

Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In *International Conference on Machine Learning*, 2021.

Eduard Gorbunov, Konstantin P Burlachenko, Zhize Li, and Peter Richtárik. Marina: Faster non-convex distributed learning with compression. In *International Conference on Machine Learning*, pages 3788–3798. PMLR, 2021.

Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. *Advances in Neural Information Processing Systems*, 2019.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021.

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

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 2020b.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International Conference on Machine Learning*, 2018.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2020.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 2017.

Ihab Mohammed, Shadha Tabatabai, Ala Al-Fuqaha, Faissal El Bouanani, Junaid Qadir, Basheer Qolomany, and Mohsen Guizani. Budgeted online selection of candidate iot clients to participate in federated learning. *IEEE Internet of Things Journal*, 2020.

Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. Fedfast: Going beyond average for faster training of federated recommender systems. In *International Conference on Knowledge Discovery & Data Mining*, 2020.

Elsa Rizk, Stefan Vlaski, and Ali H Sayed. Optimal importance sampling for federated learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

Navjot Singh, Deepesh Data, Jemin George, and Suhas Diggavi. Sparq-sgd: Event-triggered and compressed communication in decentralized stochastic optimization. *arXiv preprint arXiv:1910.14280*, 2019.

Sebastian U Stich. Local sgd converges fast and communicates little. In *International Conference on Learning Representations*, 2018.

Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms. *Journal of Machine Learning Research*, 22:1–50, 2021.

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

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *International Conference on Machine Learning*, 2015.

Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. *arXiv preprint arXiv:1708.01012*, 2017.