Heterogeneity-Guided Client Sampling: Towards Fast and Efficient Non-IID Federated Learning

Huancheng Chen
University of Texas at Austin
huanchengch@utexas.edu

Haris Vikalo
University of Texas at Austin
hvikalo@ece.utexas.edu

Abstract

Statistical heterogeneity of data present at client devices in a federated learning (FL) system renders the training of a global model in such systems difficult. Particularly challenging are the settings where due to communication resource constraints only a small fraction of clients can participate in any given round of FL. Recent approaches to training a global model in FL systems with non-IID data have focused on developing client selection methods that aim to sample clients with more informative updates of the model. However, existing client selection techniques either introduce significant computation overhead or perform well only in the scenarios where clients have data with similar heterogeneity profiles. In this paper, we propose HiCS-FL (Federated Learning via Hierarchical Clustered Sampling), a novel client selection method in which the server estimates statistical heterogeneity of a client's data using the client's update of the network's output layer and relies on this information to cluster and sample the clients. We analyze the ability of the proposed techniques to compare heterogeneity of different datasets, and characterize convergence of the training process that deploys the introduced client selection method. Extensive experimental results demonstrate that in non-IID settings HiCS-FL achieves faster convergence than state-of-the-art FL client selection schemes. Notably, HiCS-FL drastically reduces computation cost compared to existing selection schemes and is adaptable to different heterogeneity scenarios.

1 Introduction

The federated learning (FL) framework enables privacy-preserving collaborative training of machine learning (ML) models across a number of devices (clients) by avoiding the need to collect private data stored at those devices. The participating clients typically experience both the system as well as statistical heterogeneity [18]. The former describes settings where client devices have varying degree of computational resources, communication bandwidth and fault tolerance, while the latter refers to the fact that the data owned by the clients may be drawn from different distributions. In this paper, we focus on FL under statistical heterogeneity and leave studies of system heterogeneity to future work.

An early FL method, FedAvg [21], performs well in the settings where the devices train on independent and identically distributed (IID) data. However, compared to the IID scenario, training on non-IID data is detrimental to the convergence speed, variance and accuracy of the learned model. This has motivated numerous studies aiming to reduce the variance and improve convergence of FL on non-IID data [6, 9, 14, 17, 19, 30].

On another note, constraints on communication resources and therefore on the number of clients that may participate in training additionally complicate implementation of FL schemes. It would be particularly unrealistic to require regular contributions to training from all the clients in a large-scale cross-device FL system. Instead, only a fraction of clients participate in any given training round; unfortunately, this further aggravates detrimental effects of statistical heterogeneity. Selecting informative clients in non-IID FL settings is an open problem that has received considerable attention

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

from the research community [8, 11, 12]. Since privacy concerns typically prohibit clients from sharing their local data label distributions, existing studies focus on estimating informativeness of a client's update by analyzing the update itself. This motivated a family of methods that rely on the norms of local updates to assign probabilities of sampling the clients [7, 23]. Aiming to enable efficient use of the available communication and computation resources, another set of methods groups clients with similar data distributions into clusters based on the similarity between clients' model updates [2, 11]. Across the board, the existing methods still struggle to deliver desired performance in an efficient manner and cannot distinguish clients with balanced data from the clients with imbalanced data.

In this paper, we consider training a neural network model for **classification tasks** via federated learning and propose a novel adaptive clustering-based sampling method for identifying and selecting informative clients. The method, referred to as Federated Learning via Hierarchical Clustered Sampling (HiCS-FL), relies on the updates of the (fully connected) output layer in the network to determine how diverse is the clients' data and, based on that, decide which clients to sample. In particular, HiCS-FL enables heterogeneity-guided client selection by utilizing general properties of the gradients of the output layer to distinguish between clients with balanced from those with imbalanced data. Unlike the Clustered Sampling strategies [11] where the clusters of clients are sampled uniformly, HiCS-FL allocates different probabilities (importance) to the clusters according to their average estimated data heterogeneity. Numerous experiments conducted on vision datasets FMNIST, CIFAR10, Mini-ImageNet and a NLP dataset THUC news demonstrate that HiCS-FL achieves significantly faster training convergence and lower variance than the competing methods. Finally, we conduct convergence analysis of HiCS-FL and discuss implications of the results.

In summary, the contributions of the paper include: (1) Analytical characterization of the correlation between local updates of the output layer and the FL clients' data label distribution, along with an efficient method for estimating data heterogeneity; (2) a novel clustering-based algorithm for heterogeneity-guided client selection; (3) extensive simulation results demonstrating HiCS-FL provides significant improvement in terms of convergence speed and variance over competing approaches; and (4) theoretical analysis of the proposed schemes.

2 Background and Related Work

Assume the cross-device federated learning setting with N clients, where client k owns private local dataset $\mathcal{B}_k$ with $|\mathcal{B}_k|$ samples. The plain vanilla FL considers the objective

$$\min_{\theta} F(\theta) \triangleq \sum_{k=1}^{N} p_k F_k(\theta), \tag{1}$$

where θ denotes parameters of the global model, $F_k(\theta)$ is the loss (empirical risk) of model θ on $\mathcal{B}_k$, and p_k denotes the weight assigned to client k, $\sum_{k=1}^{N} p_k = 1$. In FedAvg, the weights are set to $p_k = |\mathcal{B}_k| \,/\, \sum_{i=1}^{N} |\mathcal{B}_i|$. In training round t, the server collects clients' model updates θ_k^t formed by training on local data and aggregates them to update global model as $\theta^{t+1} = \sum_{k=1}^{N} p_k \theta_k^t$.

When an FL system operates under resource constraints, typically only $K \ll N$ clients are selected to participate in any given round of training; denote the set of clients selected in round t by $\mathcal{S}^t$. In departure from FedAvg, FedProx [19] proposes an alternative strategy for sampling clients based on a multinomial distribution where the probability of selecting a client is proportional to the size of its local dataset; the global model is then formed as the average of the collected local models $\theta^{t+1} = \frac{1}{K} \sum_{k \in \mathcal{S}^t} \theta_k^t$. This sampling strategy is *unbiased* since the the updated global model is on expectation equal to the one obtained by the framework with full client participation as Eq.1.

AFL [12] is the first study to utilize local validation loss as a *value* function for computing client sampling probabilities; Power-of-Choice [8] takes a step further to propose a greedy approach to sampling clients with the largest local loss. Both of these methods require all clients to compute the local validation loss, which is often unrealistic. To address this problem, FedCor [28] models the local loss by a Gaussian Process (GP), estimates the GP parameters from experiments, and uses the GP model to predict clients' local losses without requiring them to perform validation. In [7], Optimal Client Sampling scheme aiming to minimize the variance of local updates by assigning sampling probabilities proportional to the Euclidean norm of the updates is proposed. The study in [23] models the progression of model's weights by an Ornstein-Uhlenbeck process and proposes a strategy, optimal under that assumption, for selecting clients with significant weight updates.

The clustering-based sampling method proposed in [11] uses cosine similarity [24] to group together clients with similar local updates, and proceeds to sample one client per cluster in attempt to avoid redundant gradient information. DivFL [2] follows the same principle of identifying representative clients but does so by constructing a submodular set and greedily selecting diverse clients. Both of these techniques are computationally expensive due to the high dimension of the gradients that they need to process.

In general, the overviewed methods either: (1) select diverse clients to reduce redundant information; or (2) select clients with a perceived significant contributions to the global model (high loss, large update or low class-imbalance). Efficient and effective client selection in FL remains an open challenge, motivating the heterogeneity-guided adaptive client selection method presented next.

3 HiCS-FL: Federated Learning via Hierarchical Clustered Sampling

Existing client sampling methods including Clustered Sampling [11] and DivFL [2] aim to select clients such that the resulting model update is an unbiased estimate of the true update (i.e., the update in the case of full client participation) while minimizing the variance

$$\left\| \frac{1}{N}\sum_{k=1}^{N} \nabla F_k(\theta^t) - \frac{1}{K}\sum_{k\in\mathcal{S}^t} \nabla F_k(\theta^t) \right\|_2^2. \qquad (2)$$

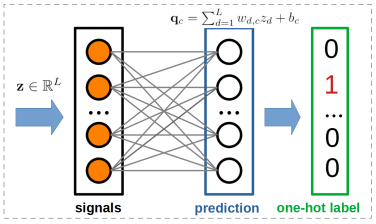


Figure 1: The last two network layers.

Clustered Sampling, for instance, groups N clients into K clusters based on *representative gradients* [24], and randomly selects one client from each cluster to contribute to the global model update. Such an approach unfortunately fails to differentiate between model updates formed on data with balanced and those formed on data with imbalanced label distributions – indeed, in either case the updates are treated as being equally important. However, a number of studies in centralized learning has shown that class-imbalanced datasets have significant detrimental effect on the performance of learning classification tasks [3, 4, 26]. This intuition carries over to the FL settings where one expects the updates from clients training on relatively more balanced local data to have a more beneficial impact on the performance of the system. The Federated Learning via Hierarchical Clustered Sampling (HiCS-FL) framework described in this section adapts to the clients' data heterogeneity in the following way: if the levels of heterogeneity (as quantified by the entropy of data label distribution) vary from one cluster to another, HiCS-FL is more likely to sample clusters containing clients with more balanced data; if the clients grouped in different clusters have similar heterogeneity levels, HiCS-FL is more likely to select diverse clients (i.e., sample uniformly across clusters, thus reducing to the conventional clustered sampling strategy).

3.1 Class-imbalance Causes Objective Drift

A number of studies explored detrimental effects of non-IID training data on the performance of a global model learned via FedAvg. An example is SCAFFOLD [14] which demonstrates *objective drift* in non-IID FL manifested through large differences between local models θ_k^* trained on substantially different data distributions. The drift is due to FedAvg updating the global model in the direction of the weighted average of local optimal models, which is not necessarily leading towards the optimal global model θ^*. The optimal model θ^*, in principle obtained by solving optimization in Eq. 1, achieves minimal empirical error on the data with uniform label distribution and is intuitively closer to the local optimal models trained on balanced data. Recent work [36] empirically verified this conjecture through extensive experiments. Let $\nabla F(\theta^t)$ denote the gradient of $F(\theta^t)$ given the global model θ^t at round t; the difference between $\nabla F(\theta^t)$ and the local gradient $\nabla F_k(\theta^t)$ computed on client k's data is typically assumed to be bounded [7, 11, 31]. To proceed, we formalize the assumption about the relationship between gradients and data label distributions.

Assumption 3.1 (Bounded Dissimilarity.) *Gradient $\nabla F_k(\theta^t)$ of the k-th local model at global round t is such that*

$$\left\|\nabla F_k(\theta^t) - \nabla F(\theta^t)\right\|^2 \leq \kappa - \rho e^{\beta\left(H(\mathcal{D}^{(k)}) - H(\mathcal{D}_0)\right)} = \sigma_k^2, \qquad (3)$$

where $\mathcal{D}^{(k)}$ is the data label distribution of client k, $\mathcal{D}_0$ denotes uniform distribution, $H(\cdot)$ is Shannon's entropy of a stochastic vector, and $\beta > 0, \kappa > \rho > 0$.

The assumption commonly encountered in literature is recovered by setting the right-hand side of (3) to $\sigma_m^2 = \max_k \sigma_k^2$. Intuitively, if the data label distribution of client k is highly imbalanced (i.e., $H(\mathcal{D}^{(k)})$ is small), the local gradient $\nabla F_k(\theta^t)$ may significantly differ from the global gradient $\nabla F(\theta^t)$ (as reflected by the bound above). Analytically, connecting the gradients to the local data label distributions allows one to characterize the effects of client selection on the variance and the rate of convergence. The results of extensive experiments that empirically verify the above assumption are reported in Appendix A.2.

3.2 Estimating Client's Data Heterogeneity

If the server were given access to clients' data label distributions, selecting clients would be relatively straightforward [32]. However, privacy concerns typically discourage clients from sharing such information. Previous studies have explored the use of multi-arm bandits for inferring clients' data heterogeneity from local model parameters, or have utilized a validation dataset at the server to accomplish the same [27, 34, 36]. In this section, we demonstrate how to efficiently and accurately estimate data heterogeneity using local updates of the output layer of a neural network in a classification task. Figure 1 illustrates the last two layers in a typical neural network. The prediction $\mathbf{q} \in \mathbb{R}^C$ is computed by forming a weighted average of signals $\mathbf{z} \in \mathbb{R}^L$ utilizing the weight matrix $\mathbf{W} \in \mathbb{R}^{C \times L}$ and bias $\mathbf{b} \in \mathbb{R}^C$.

3.2.1 Local updates of the output layer

An empirical investigation of the gradients of the output layer's weights while training with FedAvg using mini-batch stochastic gradient descent (SGD) as an optimizer is reported in [5, 29]. There, the focus is on detecting the presence of specific labels in a batch rather than on exploring the effects of class imbalance on the local update. To pursue the latter, we focus on the correlation between local updates of the output layer's bias and the client's data label distribution; we start by analyzing the training via FedAvg that employs SGD and then extend the results to other FL algorithms that utilize optimizers beyond SGD. We assume that the model is trained by minimizing the cross-entropy (CE) loss over one-hot labels – a widely used multi-class classification framework. The gradient is computed by averaging contributions of the samples in mini-batches, i.e., $\nabla_{\mathbf{b}} \mathcal{L}_{\mathbf{ce}} = \frac{1}{Bl} \sum_{j=1}^{l} \sum_{n=1}^{B} \nabla_{\mathbf{b}} \mathcal{L}_{\mathbf{ce}}^{(j,n)}(\mathbf{x}^{(j,n)}, y^{(j,n)})$, where B denotes the batch size, l is the number of mini-batches, $\mathbf{x}^{(j,n)}$ is the n-th point in the j-th mini-batch and $y^{(j,n)} \in [C]$ is its label. The contribution of $\mathbf{x}^{(j,n)}$ to the i-th component of the gradient of the output layer's bias $\mathbf{b}$ can be found as (details provided in Appendix A.3)

$$\nabla_{b_i} \mathcal{L}_{\mathbf{ce}}^{(j,n)}(\mathbf{x}^{(j,n)}, y^{(j,n)}) = \mathbb{I}\{i = y^{(j,n)}\} \frac{-\sum_{c \neq i} \exp(q_c^{(j,n)})}{\sum_{c=1}^{C} \exp(q_c^{(j,n)})} + \mathbb{I}\{i \neq y^{(j,n)}\} \frac{\exp(q_i^{(j,n)})}{\sum_{c=1}^{C} \exp(q_c^{(j,n)})}, \tag{4}$$

where $\mathbb{I}\{\cdot\}$ is an indicator, $\mathbf{q}^{(j,n)} = \mathbf{W} \cdot \mathbf{z}^{(j,n)} + \mathbf{b}$ is the output logit for signals $\mathbf{z}^{(j,n)} \in \mathbb{R}^L$ corresponding to training point $(\mathbf{x}^{(j,n)}, y^{(j,n)})$ (see Fig. 1), and where C denotes the number of classes. We make the following observations: (1) the sign of $y^{(j,n)}$-th component of $\nabla_{\mathbf{b}} \mathcal{L}_{\mathbf{ce}}^{(j,n)}$ is opposite of the sign of other components; and (2) the $y^{(j,n)}$-th component of $\nabla_{\mathbf{b}} \mathcal{L}_{\mathbf{ce}}^{(j,n)}$ is equal in magnitude to all the other components combined. Note that the above two observations are standard for neural networks using CE loss for supervised multi-class classification tasks.

In each global round t of FedAvg, the selected client k starts from the global model θ^t and proceeds to compute local update in R local epochs employing an SGD optimizer with learning rate η. According to Eq. 4, the i-th component of local update $\Delta \mathbf{b}^{(k)}$ is computed as

$$\Delta b_i^{(k)} = -\frac{\eta}{Bl} \sum_{j=1}^{l} \sum_{n=1}^{B} \sum_{r=1}^{R} \nabla_{b_i} \mathcal{L}_{\mathbf{ce}}^{(j,n,r)}, \tag{5}$$

where $\nabla_{b_i} \mathcal{L}_{\mathbf{ce}}^{(j,n,r)}$ denotes the gradient of bias at local epoch r. Note that the local update of client k, $\Delta \mathbf{b}^{(k)}$, is dependent on the label distribution of client k's data, $\mathcal{D}^{(k)} = [D_1^{(k)}, \ldots, D_C^{(k)}]^T$ and the label-specific components of $\mathbf{q}^{(j,n)}$ which change during training. We proceed by relating expected local updates to the label distributions; for convenience, we first introduce the following definition.

Definition 3.2 *Let $\mathcal{B}^{-i}$ be the subset of local data $\mathcal{B}$ that excludes points with label i. Let $\boldsymbol{s}^{-i}(\mathbf{x}) \in [0,1]^C$ be the softmax output of a trained neural network for a training point $(\mathbf{x}, y) \in \mathcal{B}^{-i}$. The i-th component of $\boldsymbol{s}^{-i}(\mathbf{x})$, $\boldsymbol{s}_i^{-i}(\mathbf{x})$, indicates the level of confidence in (erroneously) classifying $\boldsymbol{x}$ as having label i. For convenience, we define $\mathcal{E}_i = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{B}^{-i}}\left[\mathbf{s}_i^{-i}(\mathbf{x})\right], \forall i \in [C]$.*

In an untrained/initialized neural network where classifier makes random predictions, $\mathcal{E}_i = 1/C$; as training proceeds, $\mathcal{E}_i$ decreases. By taking expectation and simplifying, we obtain (details provided in Appendix A.4)

$$\mathbb{E}\left[\Delta b_i^{(k)}\right] = \eta R\left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i\right), \tag{6}$$

where $D_i^{(k)}$ denotes the true fraction of samples with label i in client k's data, $\sum_{i=1}^{C} D_i^{(k)} = 1$.

3.2.2 Estimating local data heterogeneity

We quantify the heterogeneity of clients' data by an entropy-like measure defined below. Let $\mathcal{D}^{(k)}$ denote the label distribution of client k's data; its entropy is defined as $H(\mathcal{D}^{(k)}) \triangleq -\sum_{i=1}^{C} D_i^{(k)} \ln D_i^{(k)} \leq \ln C$. Recall that more balanced data results in higher entropy, and that $H(\mathcal{D}^{(k)})$ takes the maximal value when $\mathcal{D}^{(k)}$ is uniform. The server does not know $\mathcal{D}^{(k)}$ and therefore cannot compute $H(\mathcal{D}^{(k)})$ directly. We define

$$\hat{H}(\mathcal{D}^{(k)}) \triangleq H(\text{softmax}(\Delta \mathbf{b}^{(k)}, T)), \tag{7}$$

here T is a scaling hyper-parameter (so-called *temperature*). Note that even though we can compute $\hat{H}(\mathcal{D}^{(k)})$ to characterize heterogeneity, $D_i^{(k)}$ and $\mathcal{E}_i$ remain unknown to the server (details in A.5).

Theorem 3.3 *Consider an FL system in which clients collaboratively train a model for a classification task over C classes. Let $\mathcal{D}^{(u)}$ and $\mathcal{D}^{(k)}$ denote data label distributions of an arbitrary pair of clients u and k, respectively. Moreover, let $\mathbf{U}$ denote the uniform distribution, and let η and R be the learning rate and the number of local epochs, respectively. Then*

$$\mathbb{E}\left[\hat{H}(\mathcal{D}^{(u)}) - \hat{H}(\mathcal{D}^{(k)})\right] \geq \frac{1}{2}\left(\frac{\eta R}{CT}\sum_{c=1}^{C}\mathcal{E}_c\right)^2 \left\|\mathcal{D}^{(k)} - \mathbf{U}\right\|_2^2 - \frac{\eta R}{T}\left\|\mathcal{D}^{(u)} - \mathbf{U}\right\|_\infty - \mathcal{C}\delta, \tag{8}$$

where $\mathcal{C} = \frac{\eta R(\eta R + C^2 T \ln C)}{C^2 T^2}$ and $\delta = \max_i \left|\frac{\sum_{c=1}^{C}\mathcal{E}_c}{C} - \mathcal{E}_i\right|$. The proof is provided in Appendix A.6.

As an illustration, consider the scenario where client u has a balanced dataset while the dataset of client k is imbalanced; then $\|\mathcal{D}^{(k)} - \mathbf{U}\|_2^2$ is relatively large compared to $\|\mathcal{D}^{(u)} - \mathbf{U}\|_\infty$. The bound in (8) also depends on δ, which is reflective of how misleading on average can a class be; small δ suggests that no class is universally misleading. As shown in Appendix A.4, during training δ gradually decreases to 0 as $\sum_{i=1}^{C}\mathcal{E}_i$ decreases to 0.

3.2.3 Generalizing beyond FedAvg and SGD

The proposed method for estimating clients' data heterogeneity relies on the properties of the gradient for the cross-entropy loss objective discussed in Section 3.2.1. However, for FL algorithms other than FedAvg, such as FedProx [19], FedDyn [1] and Moon [16], which add regularization to combat overfitting, the aforementioned properties may not hold. Moreover, optimization algorithms using second-order momentum such as Adam [15] deploy update rules different from SGD, making the local updates no longer proportional to the gradients. Nevertheless, HiCS-FL remains capable of distinguishing between clients with imbalanced and balanced data, which will be demonstrated in our experiments. Further theoretical discussion of various FL algorithms with optimizers beyond SGD are in appendix A.8 and A.9.

3.3 Heterogeneity-guided Clustering

Clustered Sampling [11] uses cosine similarity [24] between gradients to quantify proximity between clients' data distributions and subsequently group them into clusters. However, cosine similarity

Algorithm 1 HiCS-FL

Input:
Datasets distributed across N clients, the number of clients to sample K, total global rounds $\mathcal{T}$.

Initialize updates of bias $\Delta\mathbf{b}^{(k)} \leftarrow \mathbf{0}\ \forall k \in [N]$, global model $\theta^t \leftarrow \theta^1$, $S_0 = [N]$.
for $t = 1, \ldots, \mathcal{T}$ **do**
  if $t \leq \lceil N/K \rceil$ **then**
    $\mathcal{S}^t \leftarrow$ randomly sample $\min(K, |S_0|)$ clients from S_0, update $S_0 \leftarrow S_0 - \mathcal{S}^t$;
  else
    estimate $\hat{H}^t(\mathcal{D}^{(k)})$ and cluster N clients into M groups based on Eq. 9;
    $\mathcal{S}^t \leftarrow \emptyset$;
    while $|\mathcal{S}^t| < K$ **do**
      sample group G_m^t according to π^t;
      sample client k in G_m^t based on $\tilde{\mathbf{p}}_m$;
      $\mathcal{S}^t \leftarrow \mathcal{S}^t \cup k$;
    end while
  end if
  for $k \in \mathcal{S}^t$ **do**
    $\theta_k^t \leftarrow$ **LocalUpdate**(θ^t), $\Delta\mathbf{b}^{(k)} \in \theta_k^t - \theta^t$
  end for
  $\theta^{t+1} \leftarrow \frac{1}{K}\sum_{k\in\mathbf{S}^t} \theta_k^t$;
  $\Delta\mathbf{b}^{(k)} \leftarrow \Delta\mathbf{b}^{(k)}, \forall k \in \mathcal{S}^t$;
end for
Output:
The global model $\theta^{\mathcal{T}+1}$

cannot help distinguish between clients with balanced and those with imbalanced datasets. Motivated by this observation, we introduce a new distance measure that incorporates estimates of data heterogeneity $\hat{H}(\mathcal{D}^{(k)})$. In particular, the proposed measure of distance between clients u and k that we use to form clusters is defined as

$$\textbf{Distance}(u,k) = \arccos\left(\frac{\Delta\mathbf{b}^{(u)} \cdot \Delta\mathbf{b}^{(k)}}{|\Delta\mathbf{b}^{(u)}| \cdot |\Delta\mathbf{b}^{(k)}|}\right) + \lambda\left|\hat{H}(\mathcal{D}^{(u)}) - \hat{H}(\mathcal{D}^{(k)})\right|, \tag{9}$$

where the first term is akin to the cosine similarity used by CS with the major difference that we compute it using only the updates of the bias in the output layer, which is much more efficient than using the weights of the entire network; λ is a pre-defined hyper-parameter (set to 10 in all our experiments). For large λ, the second term dominates when there are clients with different levels of statistical heterogeneity; this allows emergence of clusters that group together clients with balanced datasets. The second term is small when clients have data with similar levels of statistical heterogeneity; in that case, the distance measure reduces to the conventional cosine similarity.

3.4 Hierarchical Clustered Sampling

To select K out of N clients in an FL system, we first organize the clients into $M \geq K$ groups via the proposed Hierarchical Clustered Sampling (HiCS) technique. In particular, during the first $\lceil N/K \rceil$ training rounds the server randomly (without replacement) selects clients and collects from them local updates of $\Delta\mathbf{b}^{(k)}$; the server then estimates $\hat{H}^t(\mathcal{D}^{(k)})$ for each selected client k and clusters the clients using the distance measure defined in Eq. 9. Let $G_1^t, \ldots, G_M^t$ denote the resulting M clusters at global round t, and let $\bar{H}_m^t = \frac{1}{|G_m|}\sum_{k\in G_m} \hat{H}^t(\mathcal{D}^{(k)})$ characterize the average heterogeneity of clients in cluster m, $m \in [M]$. Having computed $\bar{H}_m^t$, HiCS selects a cluster according to the probability vector π^t, and then from the selected cluster selects a client according to the probability vector $\tilde{\mathbf{p}}_m^t$. The two probability vectors π^t and $\tilde{\mathbf{p}}_m^t$ are defined as

$$\pi^t = \left[\frac{\exp(\gamma^t \bar{H}_1^t)}{\sum_{m=1}^M \exp(\gamma^t \bar{H}_m^t)}, \ldots, \frac{\exp(\gamma^t \bar{H}_M^t)}{\sum_{m=1}^M \exp(\gamma^t \bar{H}_m^t)}\right], \tilde{\mathbf{p}}_m^t = \left[\frac{p_{k_1}}{\sum_{k\in G_m} p_k}, \ldots, \frac{p_{k_{|G_m|}}}{\sum_{k\in G_m} p_k}\right], \tag{10}$$

where $k_1, \ldots, k_{|G_m|}$ are the indices of clients in cluster G_m, $\gamma^t = \gamma^0(1 - \frac{t}{\mathcal{T}})$ denotes an annealing hyper-parameter, and $\mathcal{T}$ is the number of global rounds. The annealing parameter is scheduled such that at first it promotes sampling clients with balanced data, thus accelerating and stabilizing the convergence of the global model. To avoid overfitting potentially caused by repeatedly selecting a small subset of clients, the annealing parameter is gradually reduced to $\gamma^t \approx 0$, when the server samples the clusters uniformly. The described procedure is formalized as Algorithm 1.

3.5 Convergence Analysis

Adopting the standard assumptions of smoothness, unbiased gradients and bounded variance [7], the following theorem holds for FedAvg with SGD optimizer.

Theorem 3.4 *Assume $F_k(\cdot)$ is L-smooth for all $k \in [N]$. Let θ^t denote parameters of the global model and let $F(\cdot)$ be defined as in Eq. 1. Furthermore, assume the stochastic gradient estimator $g_k(\theta^t)$ is unbiased and the variance is bounded such that $\mathbb{E}\left\|g_k(\theta^t) - \nabla F_k(\theta^t)\right\|^2 \leq \sigma^2$. Let η and R be the learning rate and the number of local epochs, respectively. If the learning rate is such that $\eta \leq \frac{1}{8LR}, R \geq 2$, then*

$$\min_{t\in[\mathcal{T}]} \left\|\nabla F(\theta^t)\right\|^2 \leq \frac{1}{\mathcal{T}} \left(\frac{F(\theta^0) - F(\theta^*)}{\mathcal{A}_1} + \mathcal{A}_2 \sum_{t=0}^{\mathcal{T}-1} \sum_{k=1}^{N} \omega_k^t \sigma_k^2 \right) + \mathbf{\Phi}, \tag{11}$$

where $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathbf{\Phi}$ are positive constants, and ω_k^t is the probability of sampling client k at round t.

Note that only the second term in the parenthesis on the right-hand side of the bound in Theorem 3.4 is related to the sampling method Π. Under Assumption 3.1,

$$\sum_{k=1}^{N} \omega_k^t \sigma_k^2 \leq \kappa - \sum_{k=1}^{N} \omega_k^t \frac{\exp\left(\beta H(\mathcal{D}^{(k)})\right)}{\exp\left(\beta H(\mathcal{D}_0)\right)} \rho = \kappa - \mathcal{H}_{\Pi}. \tag{12}$$

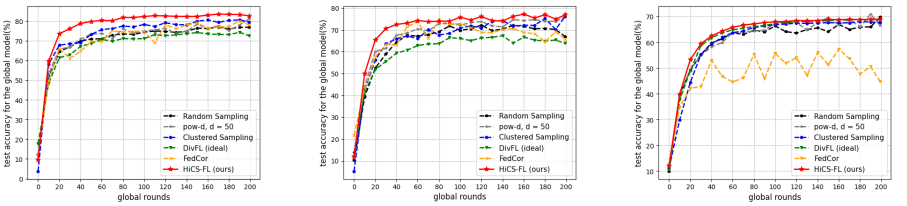
If the server samples clients with weights proportional to p_k, the statistical heterogeneity of the entire FL system may be characterized by $\mathcal{H}_{\mathbf{S}} = \sum_{k=1}^{N} p_k \frac{\exp(\beta(H(\mathcal{D}^{(k)}))}{\exp(\beta(H(\mathcal{D}_0))} \rho$. If all clients have class-imbalanced data, $\mathcal{H}_{\mathbf{S}}$ is small and thus random sampling leads to unsatisfactory convergence rate (as indicated by Theorem 3.4). On the other hand, since the clients sharing a cluster have similar data entropy, the proposed HiCS-FL leads to $\omega_k^t = \frac{p_k \exp(\gamma^t \hat{H}^t(\mathcal{D}^{(k)}))}{\sum_{j=1}^{N} p_j \exp(\gamma^t \hat{H}^t(\mathcal{D}^{(j)}))}$. When training starts, $\mathcal{H}_{\Pi}$ is large because the server tends to sample clients with higher $p_k \exp(\gamma^t H(\mathcal{D}^{(k)}))$; as γ^t decreases, $\mathcal{H}_{\Pi}$ eventually approaches $\mathcal{H}_{\mathbf{S}}$. Further details and the proof of the theorem are in Appendix A.7.

4 Experiments

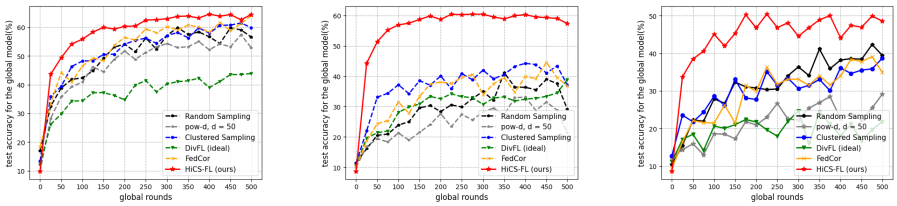
Setup. We evaluate the proposed HiCS-FL algorithm on four benchmark datasets (FMNIST, CIFAR10, Mini-ImageNet and THUC news) using different model architectures. We use four baselines: random sampling, pow-d [8], clustered sampling (CS) [11], DivFL [2] and FedCor [28]. To generate non-IID data partitions, we follow the strategy in [35], utilizing Dirichlet distribution with different concentration parameters α which controls the level of heterogeneity (smaller α leads to generating less balanced data). In a departure from previous works we utilize several different α to generate data partitions for a single experiment, leading to a realistic scenario of varied data heterogeneity across different clients. To quantify the performance of the tested methods, we use two metrics: (1) average training loss, and (2) test accuracy of the learned global model. For better visualization, data points in the results are smoothened by a Savitzky–Golay filter with window length 13 and the polynomial order set to 3. Further details of the experimental setting and a visualization of data partitions are in Appendix A.1 and A.10.

4.1 Comparison on Test Accuracy and Training Loss

FMNIST. We run FedAvg with **SGD** to train a global model which has CNN architecture in an FL system with 50 clients, where 10% of clients are selected to participate in each round of training. The data partitions are generated using one of 3 sets of the concentration parameter α values: (1) $\{0.001, 0.002, 0.005, 0.01, 0.5\}$; (2) $\{0.001, 0.002, 0.005, 0.01, 0.2\}$; (3) $\{0.001\}$. These are used to generate clients' data so as to emulate the following scenarios: (1) 80% of clients have severely imbalanced data while the remaining 20% have balanced data; (2) 80% clients have severely imbalanced data while the remaining 20% have mildly imbalanced data; (3) all clients have severely imbalanced data. Note that $\mathcal{H}_{\mathbf{M}}$ monotonically decreases as we go through settings (1) to (3). For a fair comparison, pow-d and DivFL are deployed with their ideal settings where the server requires all clients to precompute in each round a metric that is then used for client selection. Figure 2 shows that HiCS-FL outperforms other methods across different settings, exhibiting the fastest convergence rates and the least amount of variance. Particularly significant is the acceleration of convergence in setting (1) where 20% of the participating clients have balanced data. Figure 3 shows that HiCS-FL is helping achieve significant reduction of training variations (as expected, see Section 3.5) as evident by a smooth loss trajectory.

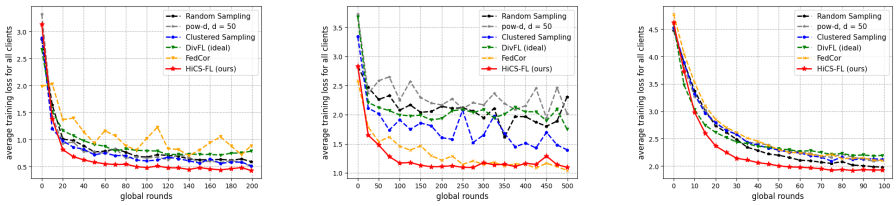


(a) FMNIST (1) (b) FMNIST (2) (c) FMNIST (3)



(d) CIFAR10 (1) (e) CIFAR10 (2) (f) CIFAR10 (3)

Figure 2: Test accuracy for the global model on 3 groups of data partitions of FMNIST and CIFAR10.



(a) FMNIST (1) (b) CIFAR10 (1) (c) Mini-ImageNet (1)

Figure 3: Training loss of HiCS-FL compared to four baselines for setting (1) on the three datasets.

CIFAR10. Here we compare the performance of HiCS-FL to FedProx [19] running CNN model with **Adam** optimizer on the task of training an FL system with 50 clients, where 20% of clients are selected to participate in each training round. Similar to the experiments on FMNIST, 3 sets of the concentration parameter α are considered: (1) $\{0.001, 0.01, 0.1, 0.5, 1\}$; (2) $\{0.001, 0.002, 0.005, 0.01, 0.5\}$; (3) $\{0.001, 0.002, 0.005, 0.01, 0.1\}$. The interpretation of the scenarios emulated by these setting is as same as in the FMNIST experiments. Figure 2 demonstrates improvement of HiCS-FL over all the other methods. HiCS-FL exhibits particularly significant improvements in settings (2) and (3), where 80% of the clients with extremely imbalanced data benefit from 20% of the clients with either balanced or mildly imbalanced data. The advantage of HiCS-FL in setting (1) where all clients have relatively high data heterogeneity is relatively modest (see Fig.2.(d)) because the system's $\mathcal{H}_{\mathbf{S}}$ is relatively large (see discussion in Section 3.5).

Mini-ImageNet. As in the Mini-ImageNet experiments, we compare HiCS-FL to FedProx running ResNet18 with **Adam** optimizer but now consider training of an FL system with 100 clients, where 20% of the clients are selected to participate in each round of training. We consider two settings of the concentration parameter α: (1) $\{0.001, 0.01, 0.1, 0.5, 1\}$ and (2) $\{0.001, 0.005, 0.01, 0.1, 1\}$. Setting (1) emulates the scenario where clients have a range of heterogeneity profiles, from extremely imbalanced, through mildly imbalanced, to balanced, while setting (2) corresponds to the scenario where 80% of the clients have extremely imbalanced data while the remaining 20% have balanced data. The system's $\mathcal{H}_{\mathbf{S}}^{(1)}$ for setting (1) is larger than $\mathcal{H}_{\mathbf{S}}^{(2)}$ for setting (2), which is reflected in a more significant improvements achieved by HiCS-FL in the latter setting, as shown in Figure 4.

THUC news. To evaluate our method on data from a different domain, we conduct experiments involving text classification on the THUC news dataset in Chinese language (10 labels). Similar to the aforementioned experiments, we allocate data to 50 clients by emulating heterogeneous data distributions scenarios with parameter α set to: (1) $\{0.001, 0.01, 0.1, 0.2,1\}$; (2) $\{0.001, 0.002, 0.01, 0.1, 0.5\}$; and (3) $\{0.001, 0.002, 0.005, 0.01, 0.1\}$. We trained TextRNNs [20] with BiLSTM architecture as

Table 1: Test accuracy (%) for the global model on 3 groups of data partitions of THUC news dataset.

Schemes	Random	Pow-of-Choice	CS	DivFL	FedCor	**HiCS-FL**
settng (1)	78.9	80.0	80.6	73.0	81.2	**83.2**
settng (2)	74.9	75.4	82.8	68.9	81.3	**83.9**
settng (3)	72.7	66.5	79.4	72.1	76.4	**79.7**

Table 2: The number of communication rounds needed to reach a certain test accuracy in the experiments on FMNIST, CIFAR10, Mini-ImageNet and THUC News. All results are for the concentration parameter setting (2).

Schemes	FMNIST		CIFAR10		Mini-ImageNet		THUC news	
	acc = 0.75	speedup	acc = 0.6	speedup	acc = 0.5	speedup	acc = 0.8	speedup
Random	149	1.0×	898	1.0×	191	1.0×	83	1.0×
pow-d	79	1.8↑	1037	0.9↓	432	0.4↓	109	0.8↓
CS	114	1.3↑	748	1.2↑	186	1.0×	74	1.1↑
DivFL	478	0.3↓	1417	0.6↓	726	0.3 ↓	289	0.3↓
FedCor	88	1.7↑	711	1.3↑	229	0.8↑	100	0.8↓
HiCS-FL	**60**	**2.5↑**	**123**	**7.3↑**	**86**	**2.2↑**	**27**	**3.1↑**

the classifiers using **Adam** optimizer. The test accuracy of the global model trained with different schemes for 100 global rounds, reported in Table 1, show that our method outperforms baselines in all the settings, demonstrating efficacy of our proposed algorithm in a simple NLP task.

4.2 Accelerating the Training Convergence

In this section we report the communication costs required to achieve convergence when using HiCS-FL, and compare those results with the competing schemes. For brevity, we select one result from each experiment conducted on the considered four datasets, and display them in Table 2. As can be seen from the table, HiCS-FL significantly reduces the number of communication rounds needed to reach target test accuracy. On FMNIST, HiCS-FL needs 60 rounds to reach test accuracy 0.75, achieving it 2.5 times faster than the random sampling scheme. On CIFAR10, HiCS-FL requires only 123 rounds to reach 0.6 test accuracy, which is 7.3 times faster than random sampling. Significant speedup appears on THUC dataset, in which HiCS-FL only needs 27 rounds to achieve 0.8 test accuracy, 3.1 times faster than the baseline. Acceleration on Mini-ImageNet is relatively modest but HiCS-FL still outperforms other methods, and does so up to 2.2 times faster than random sampling.

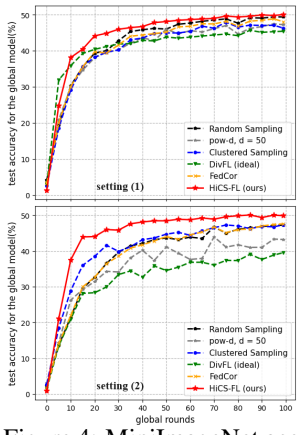


Figure 4: MiniImageNet acc.

Table 2 also shows that HiCS-FL provides the reported improvements without introducing major computational and communication overhead. The only additional computation is due to estimating data heterogeneity and performing clustering utilizing bias updates, which scales with the total number of classes but does not increase with the size of the neural network model $|\theta^t|$. Remarkably, HiCS-FL outperforms pow-d, Clustered Sampling, DivFL and FedCor in terms of convergence speed, variance and test accuracy while requiring significantly less computations. More details are provided in Appendix A.11.

4.3 Number of Clustering Groups

As discussed at the end of Section 3.3, the distance function in Equation 9 can be reduced to the conventional cosine similarity when clients exhibit similar levels of statistical heterogeneity, despite potential differences in data distribution. Under these circumstances, our HiCS-FL method can recover the performance of the previously established CS approach [11]. While CS suggests that the number of clusters M should be greater than or equal to the number of selected clients K, our HiCS-FL does not require $M > K$ but adheres to the CS settings to ensure a fair comparison. To elucidate the impact of the number of clusters, we conducted supplementary experiments with

Table 3: Additional experimental results (accuracy in %) on HiCS with the number of clusters $M \leq K$, where K is the number of selected clients each global round.

M	CIFAR10 (1)	CIFAR10 (2)	CIFAR10 (3)	Mini-ImageNet (1)	Mini-ImageNet (2)
$M = 0.3K$	61.3	57.0	47.5	50.4	50.1
$M = 0.5K$	65.1	**61.5**	46.2	49.8	50.4
$M = 0.7K$	62.8	59.2	**51.2**	**51.1**	49.9
$M = K$	**65.5**	59.8	50.6	50.5	**51.2**

Table 4: In experiments on CIFAR10, only 20 out of 50 clients are available in the beginning; additional 10 clients join each 100 global rounds. The initial 20 clients leave the system after 400 global rounds.

Scheme	Random	pow-d	DivFL	CS	FedCor	HiCS-FL
CIFAR10 (1)	85.6	86.7	84.0	86.2	80.8	**87.4**
CIFAR10 (2)	93.7	93.3	91.6	93.7	93.7	**94.7**
CIFAR10 (3)	94.5	94.7	93.9	94.5	95.0	**95.8**
Mini-ImageNet (1)	67.3	67.2	67.5	67.8	68.7	**69.0**
Mini-ImageNet (2)	71.2	71.8	72.1	72.1	**72.7**	72.5

HiCS-FL using varying numbers of clusters M and compared these results to those obtained with $M = K$ as presented in the paper. The results of those experiments can be found in Table. 3. As shown there, HiCS-FL can perform well with smaller $M < K$ as long as M is not too small, such as $M = 3$.

4.4 Dynamic Availability of Clients

The purpose of the *warm-up* phase ($t < \lceil N/K \rceil$) shown in Alg. 1 is to collect updates of the output layer from all the available clients in the system in order to facilitate clustering. Although we conduct all the experiments in the setting where clients have fixed availability, our HiCS-FL does not assume all the clients are available in the warm-up phase and can be adapted to more practical scenarios where clients have dynamic availability.

In such a scenario, the warm-up phase can be implemented by the available clients at the beginning of training. The proposed HiCS-FL is then implemented only among the available clients; the available clients with more balanced data are preferred. When new clients join the system at the global round t, the server can obtain the information of availability and selects these new clients at round $t + 1$ to approximate their data heterogeneity. To provide more insights, we conduct additional experiments on CIFAR10 dataset; the results are reported in Table. 4. As can be seen there, HiCS-FL outperforms baselines that consider clients' availability.

5 Conclusion

In this paper, we studied federated learning systems where clients that own non-IID data collaboratively train a global model; the system operates under communication constraints and thus only a fraction of clients participates in any given round of training. We developed HiCS-FL, a hierarchical clustered sampling method which estimates clients' data heterogeneity and uses this information to cluster and select clients to participate in training. We analyzed the performance of the proposed heterogeneity estimation method, and the convergence of training a FL system that deploys HiCS-FL. Extensive benchmarking experiments on four datasets demonstrated significant benefits of the proposed method, including improvement in convergence speed, variance and test accuracy, accomplished with only a minor computational overhead.

Acknowledgement

This work was funded in part by the NSF grant 2148224.

References

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

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

[3] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259.

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

[5] Huancheng Chen and Haris Vikalo. 2024. Recovering labels from local updates in federated learning. *arXiv preprint arXiv:2405.00955*.

[6] Huancheng Chen, Chaining Wang, and Haris Vikalo. 2023. The best of both worlds: Accurate global and personalized models through federated learning with data-free hyper-knowledge distillation. In *The Eleventh International Conference on Learning Representations*.

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

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

[9] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pages 2089–2099. PMLR.

[10] Sever S Dragomir, Marcel L Scholz, and Jadranka Sunde. 2000. Some upper bounds for relative entropy and applications. *Computers & Mathematics with Applications*, 39(9-10):91–100.

[11] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. 2021. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In *International Conference on Machine Learning*, pages 3407–3416. PMLR.

[12] Jack Goetz, Kshitiz Malik, Duc Bui, Seungwhan Moon, Honglei Liu, and Anuj Kumar. 2019. Active federated learning. *arXiv preprint arXiv:1909.12641*.

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

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

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[16] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10713–10722.

[17] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR.

[18] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60.

[19] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450.

[20] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

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

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

[23] Monica Ribero and Haris Vikalo. 2020. Communication-efficient federated learning via optimal client sampling. *arXiv preprint arXiv:2007.15197*.

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

[25] Ronald W Schafer. 2011. What is a savitzky-golay filter?[lecture notes]. *IEEE Signal processing magazine*, 28(4):111–117.

[26] Li Shen, Zhouchen Lin, and Qingming Huang. 2016. Relay backpropagation for effective learning of deep convolutional neural networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 467–482. Springer.

[27] Fang Shi, Weiwei Lin, Lisheng Fan, Xiazhi Lai, and Xiumin Wang. 2023. Efficient client selection based on contextual combinatorial multi-arm bandits. *IEEE Transactions on Wireless Communications*.

[28] Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen. 2022. Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10102–10111.

[29] Aidmar Wainakh, Fabrizio Ventola, Till Müßig, Jens Keim, Carlos Garcia Cordero, Ephraim Zimmer, Tim Grube, Kristian Kersting, and Max Mühlhäuser. 2021. User label leakage from gradients in federated learning. *arXiv preprint arXiv:2105.09369*.

[30] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*.

[31] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623.

[32] Joel Wolfrath, Nikhil Sreekumar, Dhruv Kumar, Yuanli Wang, and Abhishek Chandra. 2022. Haccs: Heterogeneity-aware clustered client selection for accelerated federated learning. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 985–995. IEEE.

[33] Haibo Yang, Minghong Fang, and Jia Liu. 2021. Achieving linear speedup with partial worker participation in non-iid federated learning. *arXiv preprint arXiv:2101.11203*.

[34] Miao Yang, Ximin Wang, Hongbin Zhu, Haifeng Wang, and Hua Qian. 2021. Federated learning with class imbalance reduction. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 2174–2178. IEEE.

[35] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR.

[36] Jianyi Zhang, Ang Li, Minxue Tang, Jingwei Sun, Xiang Chen, Fan Zhang, Changyou Chen, Yiran Chen, and Hai Li. 2022. Fed-cbs: A heterogeneity-aware client sampling mechanism for federated learning via class-imbalance reduction. *arXiv preprint arXiv:2209.15245*.

A Appendix

A.1 Details of the Experiments

A.1.1 General Settings

The experimental results were obtained using Pytorch [22]. In the experiments involving FMNIST, each client used a CNN-based classifier with two 5×5-convolutional layers and two 2×2-maxpooling layers (with a stride of 2), followed by a fully-connected layer. In the experiments involving CIFAR10, each client used a CNN-based classifier with three 3×3-convolutional layers and two 2×2-maxpooling layers (with a stride of 2), followed by two fully-connected layers; dimension of the hidden layer was 64. In the experiments involving Mini-ImageNet and THUC news, each client fine-tuned a pretrained ResNet18 [13] and learned a TextRNNs [20], respectively. The optimizers used for model training in the experiments on FMNIST and CIFAR10/Mini-ImageNet/THUC news were the mini-batch stochastic gradient descent (SGD) and Adam [15], respectively. The learning rate was initially set to 0.001 and then decreased every 10 iterations, with a decay factor 0.5. The number of global communication rounds was set to 200, 500, 100 and 100 for the experiments on FMNIST, CIFAR10, Mini-ImageNet and THUC news, respectively. In all the experiments, the number of local epochs R was set to 2 and the size of a mini-batch was set to 64. The sampling rate (fraction of the clients participating in a training round) was set to 0.1 for the experiments on FMNIST/THUC news, and to 0.2 for the experiments on CIFAR10/Mini-ImageNet. For the sake of visualization, data points in the presented graphs were smoothened by a Savitzky–Golay filter [25] with window length 13 and the polynomial order set to 3.

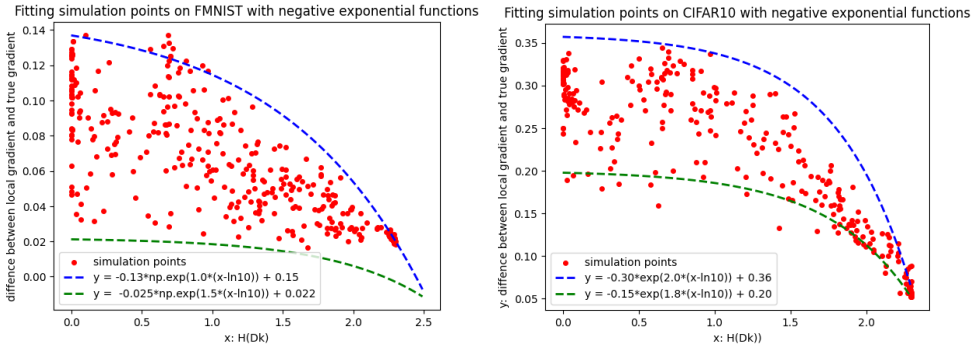
A.1.2 Hyper-parameters

In all experiments, the hyper-parameter μ of the regularization term in FedProx [19] was set to 0.1. In the Power-of-Choice (pow-d) [8] selection strategy, d was set to the total number of clients: 50 in the experiments on FMNIST, CIFAR10 and THUC news, 100 in the experiments on Mini-ImageNet. When running DivFL [2], we used the ideal setting where 1-step gradients were requested from all client in each round (regardless of their participation status), similar to the Power-of-Choice settings. For FedCor [28], we followed all settings in the paper and set the annealing coefficient β controlling the sampling strategy to 0.9 as suggested in the paper. For HiCS-FL (our method), the scaling parameter T (temperature) used in data heterogeneity estimation was set to 0.0025 in the experiments on FMNIST and to 0.0015 in the experiments on CIFAR10/Mini-ImageNet. In all experiments, parameter λ which multiplies the difference between clients' estimated data heterogeneity (used in clustering) was set to 10. In all experiments, the number of clusters m was for convenience set to be equal to the number of selected clients K. The coefficient γ^0 was set to 4 in the experiments on FMNIST and CIFAR10 while set to 2 in the experiments on Mini-ImageNet. To group clients, both Clustered Sampling [11] and HiCS-FL (our method) utilized an off-the-shelf clustering algorithm performing hierarchical clustering with Ward's Method.

A.2 Empirical Validation of Assumption 3.1

To illustrate and empirically validate Assumption 3.1, we conducted extensive experiments on FMNIST and CIFAR10 with the same model mentioned in Section A.1. In particular, we varied α over 250 values in the interval $[0.01, 50]$ to generate data partitions allocated to 250 clients; entropy of the generated label distributions ranged from 0 to $\ln 10$ (maximum). In these experiments, we allowed all clients to participate in each of 500 training rounds. To facilitate the desired study, in addition to these 250 clients we also simulated a super-client which owns a data set aggregating the data from all the clients (the set of labels in the aggregated dataset is uniformly distributed). In each round, clients start from the initialized global model and compute local gradients on their datasets; the super-client does the same on the aggregated dataset. The server computes and records squared Euclidean norm of the difference between the local gradients and the "true" gradient (i.e., the super-client's gradient). In each round, the difference between the local gradient and the true gradient changes in a pattern similar to what is stated in Assumption 3.1. As an illustration, we plot all such gradient differences computed during the entire training process of a client. Specifically, the server computes the difference between local gradient and the true gradient in each round of training, obtaining $250 \times 500 = 12500$ data points that correspond to 250 data partitions. For better visualization, we merged adjacent points.

The results obtained by following these steps in experiments on FMNIST and CIFAR10 are shown in Figure 5. For a more informative visualization, the horizontal coordinate of a point in the scatter plot is $H(\mathcal{D}^{(k)})$, while the vertical coordinate is $\|\eta_t \nabla F_k(\theta^t) - \eta_t \nabla F(\theta^t)\|^2$. The dashed lines correspond to the curves $y = -\exp(\beta\,[x - H(\mathcal{D}_0)])\rho + \kappa$ that envelop the majority of the generated points. In the case of FMNIST, the blue dashed line is parametrized by $\beta = 1.0$, $\rho = 0.13$, and $\kappa = 0.14$ while the green dashed line is parametrized by $\beta = 1.5$, $\rho = 0.025$, and $\kappa = 0.022$; these two lines envelop 95% of the generated points. In the case of CIFAR10, the blue dashed line is parametrized by $\beta = 2.0$, $\rho = 0.30$, and $\kappa = 0.36$ while the green dashed line is parametrized by $\beta = 1.8$, $\rho = 0.15$, and $\kappa = 0.20$; as in the other plot, these two lines envelop 95% of the generated points. As the plots indicate, the difference between the local gradient and the true gradient increases as $H(\mathcal{D}^{(k)})$ decreases, implying that the local gradient computed by a client with more balanced data is closer to the true gradient.



(a) FMNIST (b) CIFAR10

Figure 5: Visualization of the difference between local gradients and the global gradient (evaluated if all the data is centrally collected).

A.3 Gradient of the output (fully connected) layer's bias

Given a batch of samples $(\mathbf{x}^{(j,n)}, y^{(j,n)})$, the cross-entropy loss is readily computed as

$$\mathcal{L}_{\mathbf{CE}} = -\frac{1}{Bl}\sum_{j=1}^{l}\sum_{n=1}^{B}\log\frac{\exp\left(q_{y^{(j,n)}}^{(j,n)}\right)}{\sum_{c=1}^{C}\exp\left(q_c^{(j,n)}\right)} = \frac{1}{Bl}\sum_{j=1}^{l}\sum_{n=1}^{B}\mathcal{L}_{\mathbf{CE}}^{(j,n)}, \quad y^{(j,n)} \in [C] \tag{13}$$

$$q_c^{(j,n)} = \sum_{d=1}^{L} w_{d,c} z_d^{(j,n)} + b_c, \tag{14}$$

where B is the batchsize; l is the number of mini-batches; C is the number of classes; d is the dimension of the hidden space; $z_d^{(j,n)}$ denotes the d-th feature in the hidden space given sample $\mathbf{x}^{(j,n)}$ in the j-th batch; $w_{d,c}$ and b_c denote the weight of $z_d^{(j,n)}$ and the bias for the neuron that outputs the probability of the class c, respectively; and $q_c^{(j,n)}$ is the corresponding output logit on class c. The gradient of the bias b_i given sample $\mathbf{x}^{(j,n)}$ can be computed by the chain rule as

$$\frac{\partial \mathcal{L}_{\mathbf{CE}}^{(j,n)}}{\partial b_i} = -\frac{\partial \mathcal{L}_{\mathbf{CE}}^{(j,n)}}{\partial Q}\cdot\frac{\partial Q}{\partial q_i^{(j,n)}}\cdot\frac{\partial q_i^{(j,n)}}{\partial b_i}, \tag{15}$$

where

$$Q = \frac{\exp\left(q_{y^{(j,n)}}^{(j,n)}\right)}{\sum_{c=1}^{C}\exp\left(q_c^{(j,n)}\right)}. \tag{16}$$

Then

$$\frac{\partial \mathcal{L}_{\mathbf{CE}}^{(j,n)}}{\partial Q} = \frac{1}{Q}, \quad \frac{\partial q_i^{(j,n)}}{\partial b_i} = 1. \tag{17}$$

If $i = y^{(j,n)}$,

$$\frac{\partial Q}{\partial q_i^{(j,n)}} = \frac{\exp\left(q_{y^{(j,n)}}^{(j,n)}\right)\left(\sum_{c=1}^{C}\exp\left(q_c^{(j,n)}\right)\right) - \exp\left(q_{y^{(j,n)}}^{(j,n)}\right)^2}{\left(\sum_{c=1}^{C}\exp\left(q_c^{(j,n)}\right)\right)^2} = \frac{Q\sum_{c\neq y^{(j,n)}}\exp\left(q_c^{(j,n)}\right)}{\sum_{c=1}^{C}\exp\left(q_c^{(j,n)}\right)}. \tag{18}$$

If $i \neq y^{(j,n)}$,

$$\frac{\partial Q}{\partial q_i^{(j,n)}} = -\frac{\exp\left(q_{y^{(j,n)}}^{(j,n)}\right)\exp\left(q_i^{(j,n)}\right)}{\left(\sum_{c=1}^{C}\exp\left(q_c^{(j,n)}\right)\right)^2} = -\frac{Q\exp\left(q_i^{(j,n)}\right)}{\sum_{c=1}^{C}\exp\left(q_c^{(j,n)}\right)}. \tag{19}$$

By plugging Eq. 18 and 19 in Eq. 15, we obtain

$$\frac{\partial \mathcal{L}_{\mathbf{CE}}^{(j,n)}}{\partial b_i} = -\frac{\sum_{c\neq y^{(j,n)}}\exp\left(q_c^{(j,n)}\right)}{\sum_{c=1}^{C}\exp\left(q_c^{(j,n)}\right)},\ \text{if } i = y^{(j,n)};\ \frac{\partial \mathcal{L}_{\mathbf{CE}}^{(j,n)}}{\partial b_i} = \frac{\exp\left(q_i^{(j,n)}\right)}{\sum_{c=1}^{C}\exp\left(q_c^{(j,n)}\right)},\ \text{if } i \neq y^{(j,n)}. \tag{20}$$

A.4 Expectation of the local update $\Delta \mathrm{b}^{(k)}$

By combining Eq. 4 and 5 and taking expectation, we obtain

$$\begin{aligned}
\mathbb{E}\left[\Delta b_i^{(k)}\right] &= -\frac{\eta}{Bl}\sum_{j=1}^{l}\sum_{n=1}^{B}\sum_{r=1}^{R}\mathbb{E}\left[\nabla_{b_i}\mathcal{L}_{\mathbf{CE}}^{(j,n,r)}\right] \\
&= \eta\sum_{r=1}^{R}\mathbb{P}\{i = y^{(j,n)}\}\mathbb{I}\{i = y^{(j,n)}\}\frac{1}{Bl}\sum_{j=1}^{l}\sum_{n=1}^{B}\frac{\sum_{c\neq i}\exp(q_c^{(j,n,r)})}{\sum_{c=1}^{C}\exp(q_c^{(j,n,r)})} \\
&\quad - \eta\sum_{r=1}^{R}\mathbb{P}\{i \neq y^{(j,n)}\}\mathbb{I}\{i \neq y^{(j,n)}\}\frac{1}{Bl}\sum_{j=1}^{l}\sum_{n=1}^{B}\frac{\exp(q_i^{(j,n,r)})}{\sum_{c=1}^{C}\exp(q_c^{(j,n,r)})} \\
&= \eta\sum_{r=1}^{R}D_i^{(k)}\sum_{c\neq i}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{B}^{-c}}\left[\frac{\exp(q_c^{(j,n,r)})}{\sum_{c=1}^{C}\exp(q_c^{(j,n,r)})}\right] \\
&\quad - \eta\sum_{r=1}^{R}(1 - D_i^{(k)})\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{B}^{-i}}\left[\frac{\exp(q_i^{(j,n,r)})}{\sum_{c=1}^{C}\exp(q_c^{(j,n,r)})}\right] \\
&= \eta\sum_{r=1}^{R}D_i^{(k)}\sum_{c\neq i}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{B}^{-c}}\left[\mathbf{s}_c^{-c}(\mathbf{x})\right] - \eta\sum_{r=1}^{R}(1 - D_i^{(k)})\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{B}^{-i}}\left[\mathbf{s}_i^{-i}(\mathbf{x})\right] \\
&= \eta R\left(D_i^{(k)}\sum_{c\neq i}\mathcal{E}_c - (1 - D_i^{(k)})\mathcal{E}_i\right) \\
&= \eta R\left(D_i^{(k)}\sum_{c=1}^{C}\mathcal{E}_c - \mathcal{E}_i\right).
\end{aligned} \tag{21}$$

Note that

$$\begin{aligned}\sum_{i=1}^{C}\mathcal{E}_i &= \sum_{i=1}^{C}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{B}^{-i}}\left[\mathbf{s}_i^{-i}(\mathbf{x})\right] \\ &= \mathbb{E}\left[\sum_{i=1}^{C}\frac{1}{C-1}\sum_{c\neq i}\frac{1}{BlD_c^{(k)}}\sum_{j=1}^{l}\sum_{n=1}^{B}\mathbb{I}\{y^{(j,n)}=c\}\frac{\exp(q_i^{(j,n)})}{\sum_{c=1}^{C}\exp(q_c^{(j,n)})}\right] \\ &= \frac{1}{C-1}\sum_{i=1}^{C}\frac{1}{BlD_i^{(k)}}\sum_{j=1}^{l}\sum_{n=1}^{B}\mathbb{P}\{y^{(j,n)}=i\}\frac{\sum_{c\neq i}\exp(q_c^{(j,n)})}{\sum_{c=1}^{C}\exp(q_c^{(j,n)})} \\ &= -\frac{C}{C-1}\frac{1}{Bl}\sum_{j=1}^{l}\sum_{n=1}^{B}\frac{\exp(q_{y^{(j,n)}}^{(j,n)})}{\sum_{c=1}^{C}\exp(q_c^{(j,n)})}+\frac{C}{C-1}\end{aligned} \tag{22}$$

A comparison to $\mathcal{L}_{\mathbf{CE}}$ in Eq. 13 reveals that as $\mathcal{L}_{\mathbf{CE}}$ decreases during training, so does $\sum_{i=1}^{C}\mathcal{E}_i$. Given an untrained/initialized neural network model, $\mathcal{E}_i^0 = 1/C$ for $\forall i \in [C]$, i.e., $\sum_{i=1}^{C}\mathcal{E}_i^0 = -\frac{1}{C-1}+\frac{C}{C-1}=1$. At global round T, if $\mathcal{L}_{\mathbf{CE}}^{*}=0$, then $\sum_{i=1}^{C}\mathcal{E}_i^T = -\frac{C}{C-1}+\frac{C}{C-1}=0$.

A.5 Privacy of $\mathcal{D}^{(k)}$

According to Eq. 6, the server is able to obtain C linear equations from each client,

$$\mathbb{E}\left[\Delta b_i^{(k)}\right] = \eta R\left(D_i^{(k)}\sum_{c=1}^{C}\mathcal{E}_c - \mathcal{E}_i\right), \text{for } \forall i \in [C], \tag{23}$$

$$\sum_{i=1}^{C}D_i^{(k)} = 1, \tag{24}$$

where C denotes the number of classes. Suppose $\mathbb{E}[\Delta b_i^{(k)}]$ are known by the server. Then $D_i^{(k)}$, the variables in the aforementioned equations, cannot be determined uniquely since there are C variables and $C+1$ equations. Therefore, the server is unable to infer clients' true data label distribution and the privacy of $\mathcal{D}^{(k)}$ is protected.

A.6 Proof of Theorem 3.3

In Section A.3 we derived an expression for the gradient of the bias in the output layer given a single sample $(\mathbf{x}^{(j,n)}, y)$ in the mini-batch. It is worthwhile making the following two observations:

- the sign of the $y^{(j,n)}$-th component of $\nabla_{\mathbf{b}}\mathcal{L}_{\mathbf{CE}}^{(j,n)}(\mathbf{x}^{(j,n)}, y^{(j,n)})$ is opposite of the sign of the other components; and
- the $y^{(j,n)}$-th component of $\nabla_{\mathbf{b}}\mathcal{L}_{\mathbf{CE}}^{(j,n)}(\mathbf{x}^{(j,n)}, y^{(j,n)})$ is equal in magnitude to all other components combined.

Proof: Let $\Delta\mathbf{b}^{(k)} = [\Delta b_1^{(k)}, \ldots, \Delta b_C^{(k)}]$ denote the local update (made by client k) of the bias in the output layer of the neural network model, and let $\mathcal{D}^{(k)} = [D_1^{(k)}, \ldots, D_C^{(k)}]$ be the (unknown) true data label distribution, $\sum_{i=1}^{C}D_i^{(k)}=1$. Assuming the learning rate η and R local epochs, the expectation of the local update of $\Delta\mathbf{b}^{(k)}$ is

$$\mathbb{E}\left[\Delta b_i^{(k)}\right] = \eta R\left(D_i^{(k)}\sum_{c=1}^{C}\mathcal{E}_c - \mathcal{E}_i\right). \tag{25}$$

Data heterogeneity can be captured via entropy, $H(\mathcal{D}^{(k)}) = -\sum_{c=1}^{C}D_i^{(k)}\ln D_i^{(k)}$, where higher $H(\mathcal{D}^{(k)})$ indicates that client k has more balanced data. However, since we do not have access to

the client's data distribution, we instead define and use as a measure of heterogeneity $\hat{H}(\mathcal{D}^{(k)}) \triangleq H(\text{softmax}(\Delta\mathbf{b}^{(k)}, T))$, where

$$\text{softmax}(\Delta\mathbf{b}^{(k)}, T)_i = \frac{\exp(\Delta b_i^{(k)}/T)}{\sum_{c=1}^{C}\exp(\Delta b_c^{(k)}/T)}, \tag{26}$$

and where T denotes the *temperature* of the softmax operator. Suppose there are two clients, u and k, with class-balanced and class-imbalanced data; let $\mathcal{D}^{(u)}$ and $\mathcal{D}^{(k)}$ denote their data label distributions, respectively, while $\hat{\mathcal{D}}^{(u)}$ and $\hat{\mathcal{D}}^{(k)}$ are computed by $\text{softmax}(\Delta\mathbf{b}^{(u)}, T)$ and $\text{softmax}(\Delta\mathbf{b}^{(k)}, T)$. Without a loss of generality, we can re-parameterize $\hat{\mathcal{D}}^{(u)}$ as

$$\hat{\mathcal{D}}^{(u)} = \epsilon\mathbf{U} + \sum_{i=1}^{C}\epsilon_i\mathbf{Z}_i, \tag{27}$$

where $\mathbf{U} = [\frac{1}{C}, \ldots, \frac{1}{C}]$ denotes uniform distribution; i-th component of $\mathbf{Z}_i$ is 1 while the remaining components are 0; ϵ and ϵ_i are all non-negative such that $\epsilon + \sum_{i=1}^{C}\epsilon_i = 1$. We can always set $\min_j \epsilon_j = 0$; otherwise, let $\epsilon^{'} = \epsilon + \min_j \epsilon_j$ and $\epsilon_i^{'} = \epsilon_i - \min_j \epsilon_j,\ \forall i \in [C]$; ϵ quantifies how close is $\hat{\mathcal{D}}^{(u)}$ to $\mathbf{U}$. Due to the concavity of entropy,

$$H(\hat{\mathcal{D}}^{(u)}) \geq \epsilon H(\mathbf{U}) + \sum_{i=1}^{C}\epsilon_i H(\mathbf{Z}_i) = \epsilon \ln C. \tag{28}$$

We will find the following lemma useful.

Lemma A.1 *For two probability vectors **p** and **q** with dimension C, the Kullback–Leibler divergence between **p** and **q** satisfies*

$$KLD(\boldsymbol{p}||\boldsymbol{q}) \geq \frac{1}{2}\left\|\mathbf{p}-\mathbf{q}\right\|_1^2, \tag{29}$$

where $\left\|\mathbf{p}-\mathbf{q}\right\|_1 = \sum_{i=1}^{C}|p_i - q_i|$.

For the proof of the lemma, please see [10]. Applying it, we obtain

$$\mathbf{KLD}(\hat{\mathcal{D}}^{(k)}||\mathbf{U}) = H(\mathbf{U}) - H(\hat{\mathcal{D}}^{(k)}) \geq \frac{1}{2}\left\|\hat{\mathcal{D}}^{(k)} - \mathbf{U}\right\|_1^2 \geq \frac{1}{2}\left\|\hat{\mathcal{D}}^{(k)} - \mathbf{U}\right\|_2^2. \tag{30}$$

Combining Eq. 28 and Eq. 30, we obtain

$$H(\hat{\mathcal{D}}^{(u)}) - H(\hat{\mathcal{D}}^{(k)}) \geq (\epsilon - 1)\ln C + \frac{1}{2}\left\|\hat{\mathcal{D}}^{(k)} - \mathbf{U}\right\|_2^2. \tag{31}$$

By taking expectations of both sides,

$$\mathbb{E}\left[H(\hat{\mathcal{D}}^{(u)}) - H(\hat{\mathcal{D}}^{(k)})\right] \geq (\mathbb{E}[\epsilon] - 1)\ln C + \frac{1}{2}\mathbb{E}\left[\left\|\hat{\mathcal{D}}^{(k)} - \mathbf{U}\right\|_2^2\right]. \tag{32}$$

Since $\left\|\hat{\mathcal{D}}^{(k)} - \mathbf{U}\right\|_2^2$ is convex (composition of the Euclidean norm and softmax), according to Jensen's inequality

$$\mathbb{E}\left[H(\hat{\mathcal{D}}^{(u)}) - H(\hat{\mathcal{D}}^{(k)})\right] \geq (\mathbb{E}[\epsilon] - 1)\ln C + \frac{1}{2}\left\|\hat{\mathcal{D}}^{(k)}(\mathbb{E}[\Delta\mathbf{b}^{(k)}]) - \mathbf{U}\right\|_2^2, \tag{33}$$

where

$$\hat{\mathcal{D}}^{(k)}(\mathbb{E}[\Delta\mathbf{b}^{(k)}])_i = \frac{\exp\left(\eta R\left(D_i^{(k)}\sum_{c=1}^{C}\mathcal{E}_c - \mathcal{E}_i\right)/T\right)}{\sum_j^C \exp\left(\eta R\left(D_j^{(k)}\sum_{c=1}^{C}\mathcal{E}_c - \mathcal{E}_j\right)/T\right)}. \tag{34}$$

Selecting T such that $\eta R\left(D_i^{(k)}\sum_{c=1}^{C}\mathcal{E}_c - \mathcal{E}_i\right)/T$ is sufficiently small and applying the first-order Taylor's expansion of e^x around 0, we obtain

$$\sum_j^C \exp\left(\eta R\left(D_j^{(k)}\sum_{c=1}^{C}\mathcal{E}_c - \mathcal{E}_j\right)/T\right) = \sum_j^C 1 + \eta R\sum_j^C\left(D_j^{(k)}\sum_{c=1}^{C}\mathcal{E}_c - \mathcal{E}_j\right)/T = C, \tag{35}$$

where $\sum_{j=1}^{C} D_j^{(k)} = 1$. This leads to a simplified $\hat{\mathcal{D}}^{(k)}(\mathbb{E}[\Delta \mathbf{b}^{(k)}])$,

$$\hat{\mathcal{D}}^{(k)}(\mathbb{E}[\Delta \mathbf{b}^{(k)}])_i = \frac{1 + \eta R \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right) / T}{C}. \tag{36}$$

Substituting Eq. 36 for the second term on the right-hand side of ineq. 33 leads to

$$\left\| \hat{\mathcal{D}}^{(k)}(\mathbb{E}[\Delta \mathbf{b}^{(k)}]) - \mathbf{U} \right\|_2^2 = \left(\frac{\eta R}{CT} \right)^2 \sum_{i=1}^{C} \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right)^2. \tag{37}$$

Now, consider

$$\hat{\mathcal{D}}^{(u)} - \mathbf{U} = (\epsilon - 1)\mathbf{U} + \sum_{i=1}^{C} \epsilon_i \mathbf{Z}_i. \tag{38}$$

Taking expectations of both sides,

$$\mathbb{E}\left[(\epsilon - 1)\mathbf{U} + \sum_{i=1}^{C} \epsilon_i \mathbf{Z}_i \right] = \mathbb{E}\left[\hat{\mathcal{D}}^{(u)} - \mathbf{U} \right] \geq \hat{\mathcal{D}}^{(u)}(\mathbb{E}[\Delta \mathbf{b}^{(u)}]) - \mathbf{U}. \tag{39}$$

The above inequality holds component-wise, so for the j-component ($\epsilon_j = 0$)

$$\mathbb{E}[\frac{1}{C}(\epsilon - 1) + \epsilon_j] = \mathbb{E}[\frac{1}{C}(\epsilon - 1)] \geq \hat{\mathcal{D}}^{(u)}(\mathbb{E}[\Delta \mathbf{b}^{(u)}])_j - \mathbf{U}_i = \frac{\eta R \left(D_j^{(u)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_j \right)}{CT}. \tag{40}$$

Therefore,

$$\mathbb{E}[\epsilon] - 1 \geq \frac{\eta R \left(D_j^{(u)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_j \right)}{T} \geq \min_i \frac{\eta R \left(D_i^{(u)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right)}{T}. \tag{41}$$

Taking absolute value of both sides yields

$$|\mathbb{E}[\epsilon] - 1| \leq \frac{\eta R}{T} \max_i \left| D_i^{(u)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right| = \frac{\eta R}{T} \max_i \left| (D_i^{(u)} - \frac{1}{C}) \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i + \frac{1}{C} \sum_{c=1}^{C} \mathcal{E}_c \right|. \tag{42}$$

By applying the triangle inequality we obtain

$$|\mathbb{E}[\epsilon] - 1| \leq \frac{\eta R}{T} \max_i \left| D_i^{(u)} - \frac{1}{C} \right| \sum_{c=1}^{C} \mathcal{E}_c + \frac{\eta R}{T} \max_i \left| \frac{1}{C} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right|. \tag{43}$$

Let $\delta = \max_i \left| \frac{1}{C} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right|$. Since $\sum_{c=1}^{C} \mathcal{E}_c \leq C \frac{1}{C} = 1$, it holds that

$$|\mathbb{E}[\epsilon] - 1| \leq \frac{\eta R}{T} \max_i \left| D_i^{(u)} - \frac{1}{C} \right| + \frac{\eta R}{T} \delta. \tag{44}$$

Furthermore, since $\mathbb{E}[\epsilon] - 1 < 0$,

$$\mathbb{E}[\epsilon] - 1 \geq -\frac{\eta R}{T} \max_i \left| D_i^{(u)} - \frac{1}{C} \right| - \frac{\eta R}{T} \delta. \tag{45}$$

Note that

$$\begin{aligned}
\left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right)^2 &= \left((D_i^{(k)} - \frac{1}{C}) \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i + \frac{1}{C} \sum_{c=1}^{C} \mathcal{E}_c \right)^2 \\
&= \left((D_i^{(k)} - \frac{1}{C}) \sum_{c=1}^{C} \mathcal{E}_c \right)^2 + \left(\frac{1}{C} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right)^2 \\
&\quad + 2 \left(\sum_{c=1}^{C} \mathcal{E}_c \right) \left(D_i^{(k)} - \frac{1}{C} \right) \left(\frac{1}{C} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right) \\
&\geq \left((D_i^{(k)} - \frac{1}{C}) \sum_{c=1}^{C} \mathcal{E}_c \right)^2 + 2 \left(\sum_{c=1}^{C} \mathcal{E}_c \right) \left(D_i^{(k)} - \frac{1}{C} \right) \left(\frac{1}{C} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right).
\end{aligned} \tag{46}$$

Therefore,

$$\begin{aligned}
\sum_{i=1}^{C}\left(D_i^{(k)}\sum_{c=1}^{C}\mathcal{E}_c-\mathcal{E}_i\right)^2 &\geq \left(\sum_{c=1}^{C}\mathcal{E}_c\right)^2\sum_{i=1}^{C}\left(D_i^{(k)}-\frac{1}{C}\right)^2 \\
&\quad+2\left(\sum_{c=1}^{C}\mathcal{E}_c\right)\sum_{i=1}^{C}\left(D_i^{(k)}-\frac{1}{C}\right)\left(\frac{1}{C}\sum_{c=1}^{C}\mathcal{E}_c-\mathcal{E}_i\right) \\
&=\left(\sum_{c=1}^{C}\mathcal{E}_c\right)^2\sum_{i=1}^{C}\left(D_i^{(k)}-\frac{1}{C}\right)^2 \\
&\quad+2\left(\sum_{c=1}^{C}\mathcal{E}_c\right)\sum_{i=1}^{C}\left(\frac{D_i^{(k)}}{C}\sum_{c=1}^{C}\mathcal{E}_c-\frac{1}{C^2}\sum_{c=1}^{C}\mathcal{E}_c+\frac{\mathcal{E}_i}{C}-D_i^{(k)}\mathcal{E}_i\right) \\
&=\left(\sum_{c=1}^{C}\mathcal{E}_c\right)^2\sum_{i=1}^{C}\left(D_i^{(k)}-\frac{1}{C}\right)^2 \\
&\quad+2\left(\sum_{c=1}^{C}\mathcal{E}_c\right)\left(\frac{1}{C}\sum_{c=1}^{C}\mathcal{E}_c-\frac{1}{C}\sum_{c=1}^{C}\mathcal{E}_c+\frac{1}{C}\sum_{i=1}^{C}\mathcal{E}_i-\sum_{i=1}^{C}D_i^{(k)}\mathcal{E}_i\right) \\
&\geq \left(\sum_{c=1}^{C}\mathcal{E}_c\right)^2\sum_{i=1}^{C}\left(D_i^{(k)}-\frac{1}{C}\right)^2+2\left(\sum_{c=1}^{C}\mathcal{E}_c\right)\left(\frac{1}{C}\sum_{c=1}^{C}\mathcal{E}_c-\max_j\mathcal{E}_j\right) \\
&\geq \left(\sum_{c=1}^{C}\mathcal{E}_c\right)^2\sum_{i=1}^{C}\left(D_i^{(k)}-\frac{1}{C}\right)^2-2\delta.
\end{aligned} \tag{47}$$

Substituting the above expression in Eq. 33, we obtain

$$\mathbb{E}\left[H(\hat{\mathcal{D}}^{(u)})-H(\hat{\mathcal{D}}^{(k)})\right]\geq-\frac{\eta R\ln C}{T}\max_j\left|D_j^{(u)}-\frac{1}{C}\right|-\frac{\eta R\ln C}{T}\delta \tag{48}$$

$$+\frac{1}{2}\left(\frac{\eta R}{CT}\right)^2\left(\sum_{c=1}^{C}\mathcal{E}_c\right)^2\sum_{i=1}^{C}\left(D_i^{(k)}-\frac{1}{C}\right)^2-\left(\frac{\eta R}{CT}\right)^2\delta, \tag{49}$$

and, therefore,

$$\mathbb{E}\left[H(\hat{\mathcal{D}}^{(u)})-H(\hat{\mathcal{D}}^{(k)})\right]\geq\frac{1}{2}\left(\frac{\eta R}{CT}\sum_{c=1}^{C}\mathcal{E}_c\right)^2\left\|\mathcal{D}^{(k)}-\mathbf{U}\right\|_2^2-\frac{\eta R\ln C}{T}\left\|\mathcal{D}^{(u)}-\mathbf{U}\right\|_\infty-\mathcal{C}\delta, \tag{50}$$

where $\mathcal{C}=\frac{\eta R(\eta R+C^2T\ln C)}{C^2T^2}$. ■

A.7 Convergence Analysis

Here we present the convergence analysis of an FL system deploying FedAvg with SGD wherein only a small fraction of clients participates in any given round of training. Recall that the objective function that comes up when training a neural network model is generally non-convex; we make the standard assumptions of smoothness, unbiased gradient estimate, and bounded variance.

Assumption A.2 (Smoothness) *Each local objective function $F_k(\cdot)$ is L-smooth,*

$$\left\|\nabla F_k(\theta_k^{t+1})-\nabla F_k(\theta_k^t)\right\|_2\leq L\left\|\theta_k^{t+1}-\theta_k^t\right\|_2. \tag{51}$$

Assumption A.3 (Gradient oracle) *The stochastic gradient estimator $g_k(\theta_k^{t,r})=\nabla F_k(\theta_k^{t,r})+\zeta_k^{t,r}$ for each global round t and local epoch r is such that*

$$\mathbb{E}[\zeta_k^{t,r}]=0 \tag{52}$$

and

$$\mathbb{E}\left[\left\|\zeta_k^{t,r}\right\|^2 |\theta_k^{t,r}\right] \leq \sigma^2. \tag{53}$$

With these three assumptions in place, we provide the proof of Theorem 3.4 stated in the main paper. The proof relies on the technique previously used in [7, 33], where the sampling method is unbiased and thus $\mathbb{E}\left[\frac{1}{K}\sum_{k\in\mathcal{S}^t}\sum_{r=1}^{R} g_k(\theta_k^{t,r})\right] = \sum_{k=1}^{N}\sum_{r=1}^{R} p_k \nabla F_k(\theta_k^{t,r})$. We provide a generalization that holds for any sampling strategy, resulting in $\mathbb{E}\left[\frac{1}{K}\sum_{k\in\mathcal{S}^t}\sum_{r=1}^{R} g_k(\theta_k^{t,r})\right] = \sum_{k=1}^{N}\sum_{r=1}^{R} \omega_k^t \nabla F_k(\theta_k^{t,r})$, where ω_k^t denotes the probability of sampling client k in round t under sampling strategy Π. Note that $\sum_{k=1}^{N}\omega_k^t = 1$. We assume that all clients deploy the same number of local epochs R and use learning rate η at round t.

A.7.1 key lemma

Lemma A.4 *(Lemma 2 in [33]) Instate Assumptions 3.1, A.2 and A.3. For any step size η such that $\eta \leq \frac{1}{8LR}$, for any client k it holds that*

$$\mathbb{E}\left[\left\|\theta_k^{t,r} - \theta^t\right\|^2\right] \leq 5R\eta^2(\sigma^2 + 6R\sigma_k^2) + 30R^2\eta^2\left\|\nabla F(\theta^t)\right\|^2. \tag{54}$$

Proof of Lemma A.4: For any client $k \in [N]$ and $r \in [R]$,

$$\begin{aligned}
\mathbb{E}\left[\left\|\theta_k^{t,r} - \theta^t\right\|^2\right] &= \mathbb{E}\left[\left\|\theta_k^{t,r-1} - \theta^t - \eta g_k(\theta_k^{t,r-1})\right\|^2\right] \\
&= \mathbb{E}[\|\theta_k^{t,r-1} - \theta^t - \eta(g_k(\theta_k^{t,r-1}) - \nabla F_k(\theta_k^{t,r-1}) + \nabla F_k(\theta_k^{t,r-1}) \\
&\quad - \nabla F_k(\theta^t) + \nabla F_k(\theta^t) - \nabla F(\theta^t) + \nabla F(\theta^t))\|^2] \\
&\leq \left(1 + \frac{1}{2R-1}\right)\mathbb{E}\left\|\theta_k^{t,r-1} - \theta^t\right\|^2 + \eta^2\mathbb{E}\left\|g_k(\theta_k^{t,r-1}) - \nabla F_k(\theta_k^{t,r-1})\right\|^2 \\
&\quad + 6R\eta^2\mathbb{E}\left\|\nabla F_k(\theta_k^{t,r-1}) - \nabla F_k(\theta^t)\right\|^2 + 6R\eta^2\mathbb{E}\left\|g_k(\nabla F_k(\theta^t) - \nabla F(\theta^t)\right\|^2 \\
&\quad + 6R\eta^2\mathbb{E}\left\|\nabla F(\theta^t)\right\|^2 \\
&\leq \left(1 + \frac{1}{2R-1}\right)\mathbb{E}\left\|\theta_k^{t,r-1} - \theta^t\right\|^2 + \eta^2\sigma^2 + 6R\eta^2L^2\mathbb{E}\left\|\theta_k^{t,r-1} - \theta^t\right\|^2 \\
&\quad + 6R\eta^2\sigma_k^2 + 6R\eta^2\mathbb{E}\left\|\nabla F(\theta^t)\right\|^2 \\
&= \left(1 + \frac{1}{2R-1} + 6R\eta^2L^2\right)\mathbb{E}\left\|\theta_k^{t,r-1} - \theta^t\right\|^2 + \eta^2\sigma^2 + 6R\eta^2\sigma_k^2 \\
&\quad + 6R\eta^2\mathbb{E}\left\|\nabla F(\theta^t)\right\|^2 \\
&\leq \left(1 + \frac{1}{R-1}\right)\mathbb{E}\left\|\theta_k^{t,r-1} - \theta^t\right\|^2 + \eta^2\sigma^2 + 6R\eta^2\sigma_k^2 + 6R\eta^2\mathbb{E}\left\|\nabla F(\theta^t)\right\|^2.
\end{aligned} \tag{55}$$

Unrolling the recursion yields

$$\begin{aligned}
\mathbb{E}\left[\left\|\theta_k^{t,r} - \theta^t\right\|^2\right] &\leq \sum_{r=1}^{R}\left(1 + \frac{1}{R-1}\right)^{r-1}\left(\eta^2\sigma^2 + 6R\eta^2\sigma_k^2 + 6R\eta^2\mathbb{E}\left\|\nabla F(\theta^t)\right\|^2\right) \\
&\leq (R-1)\left[\left(1 + \frac{1}{R-1}\right)^R - 1\right]\left(\eta^2\sigma^2 + 6R\eta^2\sigma_k^2 + 6R\eta^2\mathbb{E}\left\|\nabla F(\theta^t)\right\|^2\right) \\
&\leq 5R\eta^2\left(\sigma^2 + 6R\sigma_k^2\right) + 30R^2\eta^2\left\|\nabla F(\theta^t)\right\|^2.
\end{aligned} \tag{56}$$

■

A.7.2 Proof of Theorem 3.4

The model update at global round t is formed as

$$\theta^{t+1} = \theta^t - \eta \frac{1}{K} \sum_{k \in \mathbf{S}^t} \sum_{r=1}^{R} g_k(\theta_k^{t,r}), \tag{57}$$

where θ^{t+1} and θ^t denote parameters of the global model at rounds $t+1$ and t, respectively, and $\theta_k^{t,r}$ denotes parameters of the local model of client k after r local training epochs. Let

$$\Delta^t \triangleq \frac{1}{K} \sum_{k \in \mathbf{S}^t} \sum_{r=1}^{R} g_k(\theta_k^{t,r}). \tag{58}$$

Taking the expectations (conditioned on θ^t) of both sides, we obtain

$$\begin{aligned}
\mathbb{E}\left[F(\theta^{t+1})\right] &= \mathbb{E}\left[F(\theta^t - \eta\Delta^t)\right] \\
&\overset{(a)}{\leq} F(\theta^t) - \eta \left\langle \nabla F(\theta^t), \mathbb{E}\left[\Delta^t\right] \right\rangle + \frac{L}{2}\eta^2 \mathbb{E}\left[\left\|\Delta^t\right\|^2\right] \\
&= F(\theta^t) + \eta \left\langle \nabla F(\theta^t), \mathbb{E}\left[R\nabla F(\theta^t) - R\nabla F(\theta^t) - \Delta^t\right] \right\rangle + \frac{L}{2}\eta^2 \mathbb{E}\left[\left\|\Delta^t\right\|^2\right] \\
&= F(\theta^t) - R\eta \left\|\nabla F(\theta^t)\right\|^2 + \eta \underbrace{\left\langle \nabla F(\theta^t), \mathbb{E}\left[R\nabla F(\theta^t) - \Delta^t\right] \right\rangle}_{A_1} + \frac{L}{2}\eta^2 \underbrace{\mathbb{E}\left[\left\|\Delta^t\right\|^2\right]}_{A_2}.
\end{aligned} \tag{59}$$

Inequality (a) in the expression above holds due to the smoothness of $F(\cdot)$ (see Assumption A.2). Note that the term A_1 can be bounded as

$$\begin{aligned}
A_1 &= \left\langle \nabla F(\theta^t), \mathbb{E}\left[R\nabla F(\theta^t) - \Delta^t\right] \right\rangle \\
&= \left\langle \nabla F(\theta^t), \mathbb{E}\left[R\nabla F(\theta^t) - \frac{1}{K}\sum_{k \in \mathcal{S}^t}\sum_{r=1}^{R} g_k(\theta_k^{t,r})\right] \right\rangle \\
&= \left\langle \nabla F(\theta^t), \mathbb{E}\left[R\nabla F(\theta^t)\right] - \sum_{k=1}^{N}\sum_{r=1}^{R} \omega_k^t \nabla F_k(\theta_k^{t,r}) \right\rangle \\
&= \sum_{k=1}^{N} \omega_k^t \left\langle \sqrt{R}\nabla F(\theta^t), -\frac{1}{\sqrt{R}}\mathbb{E}\left[\sum_{r=1}^{R}\left(\nabla F_k(\theta_k^{t,r}) - \nabla F(\theta^t)\right)\right] \right\rangle \\
&\overset{(a)}{=} \frac{R}{2}\left\|\nabla F(\theta^t)\right\|^2 + \frac{1}{2R}\sum_{k=1}^{N}\omega_k^t \mathbb{E}\left\|\sum_{r=1}^{R}\left(\nabla F_k(\theta_k^{t,r}) - \nabla F(\theta^t)\right)\right\|^2 - \frac{1}{2R}\sum_{k=1}^{N}\omega_k^t \mathbb{E}\left\|\sum_{r=1}^{R}\nabla F_k(\theta_k^{t,r})\right\|^2 \\
&\overset{(b)}{\leq} \frac{R}{2}\left\|\nabla F(\theta^t)\right\|^2 + \frac{1}{R}\sum_{k=1}^{N}\omega_k^t \mathbb{E}\left\|\sum_{r=1}^{R}\left(\nabla F_k(\theta_k^{t,r}) - \nabla F_k(\theta^t)\right)\right\|^2 \\
&\quad + \frac{1}{R}\sum_{k=1}^{N}\omega_k^t \mathbb{E}\left\|\sum_{r=1}^{R}\left(\nabla F_k(\theta^t) - \nabla F(\theta^t)\right)\right\|^2 - \frac{1}{2R}\sum_{k=1}^{N}\omega_k^t \mathbb{E}\left\|\sum_{r=1}^{R}\nabla F_k(\theta_k^{t,r})\right\|^2 \\
&\overset{(c)}{\leq} \frac{R}{2}\left\|\nabla F(\theta^t)\right\|^2 + \sum_{k=1}^{N}\omega_k^t \sum_{r=1}^{R}\mathbb{E}\left\|\nabla F_k(\theta_k^{t,r}) - \nabla F_k(\theta^t)\right\|^2 \\
&\quad + \sum_{k=1}^{N}\omega_k^t \sum_{r=1}^{R}\mathbb{E}\left\|\nabla F_k(\theta^t) - \nabla F(\theta^t)\right\|^2 - \frac{1}{2R}\sum_{k=1}^{N}\omega_k^t \mathbb{E}\left\|\sum_{r=1}^{R}\nabla F_k(\theta_k^{t,r})\right\|^2 \\
&\overset{(d)}{\leq} \frac{R}{2}\left\|\nabla F(\theta^t)\right\|^2 + L^2\sum_{k=1}^{N}\omega_k^t \sum_{r=1}^{R}\mathbb{E}\left\|\theta_k^{t,r} - \theta^t\right\|^2 + R\sum_{k=1}^{N}\omega_k^t\sigma_k^2 - \frac{1}{2R}\sum_{k=1}^{N}\omega_k^t \mathbb{E}\left\|\sum_{r=1}^{R}\nabla F_k(\theta_k^{t,r})\right\|^2,
\end{aligned} \tag{60}$$

where equality (a) follows from $\langle \mathbf{x}, \mathbf{y} \rangle = \frac{1}{2}\left(\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \|\mathbf{x}-\mathbf{y}\|^2\right)$, inequality (b) is due to $\|\mathbf{x}+\mathbf{y}\|^2 \leq 2\|\mathbf{x}\|^2 + 2\|\mathbf{y}\|^2$, inequality (c) holds because $\left\|\sum_{i=1}^n \mathbf{z}_i\right\|^2 \leq n\sum_{i=1}^n \|\mathbf{z}_i\|^2$, and inequality (d) follows from Assumptions 3.1 and A.2. By selecting $\eta < \frac{1}{8LR}$ and applying Lemma A.4 we obtain

$$\begin{aligned}
A_1 &\leq \frac{R}{2}\left\|\nabla F(\theta^t)\right\|^2 + L^2 \sum_{k=1}^N \omega_k^t \sum_{r=1}^R \left[5R\eta^2(\sigma^2 + 6R\sigma_k^2) + 30R^2\eta^2\left\|\nabla F(\theta^t)\right\|^2\right] \\
&\quad + R\sum_{k=1}^N \omega_k^t \sigma_k^2 - \frac{1}{2R}\sum_{k=1}^N \omega_k^t \mathbb{E}\left\|\sum_{r=1}^R \nabla F_k(\theta_k^{t,r})\right\|^2 \\
&= \left(\frac{R}{2} + 30L^2R^3\eta^2\right)\left\|\nabla F(\theta^t)\right\|^2 + 5L^2R^2\eta^2\sigma^2 + 30L^2R^3\eta^2\sum_{k=1}^N \omega_k^t\sigma_k^2 \\
&\quad + R\sum_{k=1}^N \omega_k^t \sigma_k^2 - \frac{1}{2R}\sum_{k=1}^N \omega_k^t \mathbb{E}\left\|\sum_{r=1}^R \nabla F_k(\theta_k^{t,r})\right\|^2.
\end{aligned} \tag{61}$$

Furthermore,

$$\begin{aligned}
A_2 &= \mathbb{E}\left[\left\|\frac{1}{K}\sum_{k\in\mathcal{S}^t}\sum_{r=1}^R g_k(\theta_k^{t,r})\right\|^2\right] \\
&= \mathbb{E}\left[\left\|\sum_{k=1}^N \frac{\mathbb{I}\{k\in\mathcal{S}^t\}}{K}\sum_{r=1}^R g_k(\theta_k^{t,r})\right\|^2\right] \\
&= \mathbb{E}\left[\left\|\sum_{k=1}^N \frac{\mathbb{I}\{k\in\mathcal{S}^t\}}{K}\sum_{r=1}^R g_k(\theta_k^{t,r}) - \nabla F_k(\theta_k^{t,r}) + \nabla F_k(\theta_k^{t,r})\right\|^2\right] \\
&\overset{(a)}{=} \mathbb{E}\left[\left\|\sum_{k=1}^N \frac{\mathbb{I}\{k\in\mathcal{S}^t\}}{K}\sum_{r=1}^R g_k(\theta_k^{t,r}) - \nabla F_k(\theta_k^{t,r})\right\|^2\right] + \mathbb{E}\left[\left\|\sum_{k=1}^N \frac{\mathbb{I}\{k\in\mathcal{S}^t\}}{K}\sum_{r=1}^R \nabla F_k(\theta_k^{t,r})\right\|^2\right] \\
&\overset{(b)}{\leq} \mathbb{E}\left[\sum_{k=1}^N \frac{\mathbb{I}\{k\in\mathcal{S}^t\}}{K}\sum_{r=1}^R \left\|g_k(\theta_k^{t,r}) - \nabla F_k(\theta_k^{t,r})\right\|^2\right] + \mathbb{E}\left[\sum_{k=1}^N \frac{\mathbb{I}\{k\in\mathcal{S}^t\}}{K}\left\|\sum_{r=1}^R \nabla F_k(\theta_k^{t,r})\right\|^2\right] \\
&\overset{(c)}{\leq} R\sigma^2 + \sum_{k=1}^N \omega_k^t \mathbb{E}\left\|\sum_{r=1}^R \nabla F_k(\theta_k^{t,r})\right\|^2,
\end{aligned} \tag{62}$$

where equation (a) holds because $\mathbb{E}\left[g_k(\theta_k^{t,r}) - \nabla F_k(\theta_k^{t,r})\right] = 0$, inequality (b) stems from the Jensen's inequality, and inequality (c) is due to Assumption A.3.

Substituting inequalities (61) and (62) into inequality (59) yields

$$\begin{aligned}
\mathbb{E}\left[F(\theta^{t+1})\right] &\leq F(\theta^t) - R\eta\left\|\nabla F(\theta^t)\right\|^2 + \eta\underbrace{\left\langle \nabla F(\theta^t), \mathbb{E}\left[R\nabla F(\theta^t) - \Delta^t\right]\right\rangle}_{A_1} + \frac{L}{2}\eta^2\underbrace{\mathbb{E}\left[\left\|\Delta^t\right\|^2\right]}_{A_2} \\
&\leq F(\theta^t) - R\eta\left(\frac{1}{2} - 30L^2R^2\eta^2\right)\left\|\nabla F(\theta^t)\right\|^2 + \left(5L^2R^2\eta^3 + \frac{LR}{2}\eta^2\right)\sigma^2 \\
&\quad + \left(30L^2R^3\eta^3 + R\eta\right)\sum_{k=1}^N \omega_k^t\sigma_k^2 + \left(\frac{L}{2}\eta^2 - \frac{\eta}{2R}\right)\sum_{k=1}^N \omega_k^t \mathbb{E}\left\|\sum_{r=1}^R \nabla F_k(\theta_k^{t,r})\right\|^2.
\end{aligned} \tag{63}$$

If $\eta < \frac{1}{8LR}$, it must be that $\frac{1}{2} - 30L^2R^2\eta^2 > 0$ and $\frac{L}{2}\eta^2 - \frac{\eta}{2R} < 0$, leading to

$$\begin{aligned}\mathbb{E}\left[F(\theta^{t+1})\right] \leq F(\theta^t) - R\eta\left(\frac{1}{2} - 30L^2R^2\eta^2\right)\left\|\nabla F(\theta^t)\right\|^2 \\ + \left(5L^2R^2\eta^3 + \frac{LR}{2}\eta^2\right)\sigma^2 + \left(30L^2R^3\eta^3 + R\eta\right)\sum_{k=1}^{N}\omega_k^t\sigma_k^2.\end{aligned} \tag{64}$$

By rearranging and summing from $t = 0$ to $t = \mathcal{T} - 1$ we obtain

$$\begin{aligned}\mathbb{E}\left[F(\theta^{\mathcal{T}})\right] - F(\theta^0) &\leq -R\eta\left(\frac{1}{2} - 30L^2R^2\eta^2\right)\sum_{t=0}^{\mathcal{T}-1}\left\|\nabla F(\theta^t)\right\|^2 \\ &\quad + \left(5L^2R^2\eta^3 + \frac{LR}{2}\eta^2\right)\mathcal{T}\sigma^2 + \left(30L^2R^3\eta^3 + R\eta\right)\sum_{t=0}^{\mathcal{T}-1}\sum_{k=1}^{N}\omega_k^t\sigma_k^2 \\ &\leq -R\eta\left(\frac{1}{2} - 30L^2R^2\eta^2\right)\mathcal{T}\min_{t\in[\mathcal{T}]}\left\|\nabla F(\theta^t)\right\|^2 \\ &\quad + \left(5L^2R^2\eta^3 + \frac{LR}{2}\eta^2\right)\mathcal{T}\sigma^2 + \left(30L^2R^3\eta^3 + R\eta\right)\sum_{t=0}^{\mathcal{T}-1}\sum_{k=1}^{N}\omega_k^t\sigma_k^2.\end{aligned} \tag{65}$$

Let θ^* denote the optimal model's parameters, i.e., $F(\theta^*) \leq F(\theta^t) \forall t \in [\mathcal{T}]$. Then

$$\min_{t\in[\mathcal{T}]}\left\|\nabla F(\theta^t)\right\|^2 \leq \frac{1}{\mathcal{T}}\left(\frac{F(\theta^0) - F(\theta^*)}{\mathcal{A}_1} + \mathcal{A}_2\sum_{t=0}^{\mathcal{T}-1}\sum_{k=1}^{N}\omega_k^t\sigma_k^2\right) + \mathbf{\Phi}, \tag{66}$$

where $\mathcal{A}_1 = R\eta\left(\frac{1}{2} - 30L^2R^2\eta^2\right)$, $\mathcal{A}_2 = \frac{60L^2R^3\eta^3 + 2R\eta}{R\eta(1-60L^2R^2\eta^2)}$ and $\mathbf{\Phi} = \frac{\left(10L^2R\eta^2 + L\eta\right)\sigma^2}{1-60L^2R^2\eta^2}$.

■

A.8 Regularization Terms in the Objective Function

The proposed method for estimating clients' data heterogeneity relies on the properties of gradient computed for the cross-entropy loss objective. However, the method also applies to the FL algorithms other than FedAvg, in particular those that add a regularization term to combat overfitting, including FedProx [19], FedDyn[1] and Moon [16]. In the following discussion, we demonstrate that HiCS-FL remains capable of distinguishing between clients with imbalanced and balanced data when using these other FL algorithms.

A.8.1 FedProx

The objective function used by FedProx [19] is

$$\mathcal{L}_{\text{prox}}^r = \mathcal{L}_{\mathbf{CE}}^r + \frac{\mu}{2}\left\|\theta_k^{t,r} - \theta^t\right\|^2, \tag{67}$$

where $\theta_k^{t,r}$ is the vector of client k's local model parameters in the r-th local epoch at global round t. Therefore, contribution of sample $(\mathbf{x}^{(j,n)}, y^{(j,n)})$ to the gradient of $\mathcal{L}_{\text{prox}}$ in local epoch r is

$$\frac{\partial\mathcal{L}_{\text{prox}}^{(j,n,r)}}{\partial b_i} = \frac{\partial\mathcal{L}_{\mathbf{CE}}^{(j,n,r)}}{\partial b_i} + \mu\left(b_i^{t,r} - b_i^t\right), \tag{68}$$

where $\mathbf{b}^{t,r} = [b_1^{t,r}, \ldots, b_C^{t,r}]$ denotes parameters of bias in the output layer of the local model, and $\mathbf{b}^t = [b_1^t, \ldots, b_C^t]$ denotes parameters of the global model at round t. We assume the model is trained by SGD as the optimizer, and hence

$$b_i^{t,r} - b_i^t = b_i^{t,r-1} - \eta_t\frac{\partial\mathcal{L}_{\text{prox}}^{(j,n,r-1)}}{\partial b_i} - b_i^t = -\eta_t\frac{\partial\mathcal{L}_{\mathbf{CE}}^{(j,n,r-1)}}{\partial b_i} + (1 - \eta_t\mu)(b_i^{t,r-1} - b_i^t). \tag{69}$$

Therefore,

$$\begin{aligned} b_i^{t,r} - b_i^t &= -\eta_t \sum_{s=1}^{r-1} (1-\eta_t\mu)^{r-1-s} \frac{\partial \mathcal{L}_{\mathbf{CE}}^{(j,n,s)}}{\partial b_i} + (1-\eta_t\mu)^{r-1}(b_i^t - b_i^t) \\ &= -\eta_t \sum_{s=1}^{r-1} (1-\eta_t\mu)^{r-1-s} \frac{\partial \mathcal{L}_{\mathbf{CE}}^{(j,n,s)}}{\partial b_i}, \end{aligned} \tag{70}$$

and thus

$$\frac{\partial \mathcal{L}_{\text{prox}}^{(j,n,r)}}{\partial b_i} = \frac{\partial \mathcal{L}_{\mathbf{CE}}^{(j,n,r)}}{\partial b_i} - \eta_t\mu \sum_{s=1}^{r-1} (1-\eta_t\mu)^{r-1-s} \frac{\partial \mathcal{L}_{\mathbf{CE}}^{(j,n,s)}}{\partial b_i}. \tag{71}$$

Taking expectation of both sides yields

$$\begin{aligned} \frac{1}{Bl} \sum_{j=1}^{l} \sum_{n=1}^{B} \sum_{r=1}^{R} \mathbb{E}\left[\frac{\partial \mathcal{L}_{\text{prox}}^{(j,n,r)}}{\partial b_i} \right] &= \left(-\mathbb{E}[\mathbb{I}\{i = y^{(j,n)}\}] \sum_{c \neq i} \mathcal{E}_c + \mathbb{E}[\mathbb{I}(i \neq y^{(j,n)})] \mathcal{E}_i \right) \\ &\quad \cdot \sum_{r=1}^{R} \left(1 - \eta_t\mu \sum_{s=1}^{r-1} (1-\eta_t\mu)^{r-1-s} \right) \\ &= \sum_{r=1}^{R} \left(-D_i^{(k)} \sum_{c \neq i} \mathcal{E}_c + (1 - D_i^{(k)}) \mathcal{E}_i \right) \left(1 - \eta_t\mu \frac{1-(1-\eta_t\mu)^{r-1}}{\eta_t\mu} \right) \\ &= \sum_{r=1}^{R} c^r \left(-D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c + \mathcal{E}_i \right), \end{aligned} \tag{72}$$

where $c^r = (1-\eta_t\mu)^{r-1} > 0$ provided η_t and μ are sufficiently small. Therefore, the expectation of the local update of bias in the output layer satisfies

$$\mathbb{E}\left[\Delta b_i^{(k)} \right] = C\eta_t \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right), \tag{73}$$

where $C = \sum_{r=1}^{R} c^r$. Eq. (73) is similar to the expression for the expectation of the local updates of bias when applying FedAvg presented in the main paper; clearly, the analysis of HiCS-FL done in the context of FedAvg extends to FedProx.

A.8.2 FedDyn

For FedDyn [1], the objective function in local epoch r at global round t is

$$\mathcal{L}_{\text{dyn}}^{t,r} = \mathcal{L}_{\mathbf{CE}}^{t,r} - \left\langle \nabla \mathcal{L}_{\text{dyn}}^{t-1,R}, \theta_k^{t,r} \right\rangle + \frac{\mu}{2} \left\| \theta_k^{t,r} - \theta^t \right\|^2, \tag{74}$$

where R denotes the total number of local epochs. The first order condition for local optima implies

$$\nabla \mathcal{L}_{\text{dyn}}^{t,r} - \nabla \mathcal{L}_{\text{dyn}}^{t-1,R} + \mu(\theta_k^{t,r} - \theta^t) = 0, \tag{75}$$

and, therefore,

$$\begin{aligned}
\frac{\partial \mathcal{L}_{\text{dyn}}^{t,r}}{\partial b_i} &= \frac{\partial \mathcal{L}_{\text{dyn}}^{t-1,R}}{\partial b_i} - \mu \left(b_i^{t,r} - b_i^t\right) \\
&= \frac{\partial \mathcal{L}_{\text{dyn}}^{t-2,R}}{\partial b_i} - \mu \left(b_i^{t-1,R} - b_i^{t-1}\right) - \mu \left(b_i^{t,r} - b_i^t\right) \\
&= -\mu \sum_{\tau=1}^{t-1} \left(b_i^{\tau,R} - b_i^{\tau}\right) - \mu \left(b_i^{t,r} - b_i^t\right) \\
&= -\mu \sum_{\tau=1}^{t-1} \Delta b_i^{\tau} - \mu \left(b_i^{t,r} - b_i^t\right) \\
&= -\mu \sum_{\tau=1}^{t-1} \Delta b_i^{\tau} - \mu \left(-\eta_t \frac{\partial \mathcal{L}_{\text{dyn}}^{t,r-1}}{\partial b_i} + b_i^{t,r-1} - b_i^t\right) \\
&= -\mu \sum_{\tau=1}^{t-1} \Delta b_i^{\tau} + \mu \eta_t \left(\sum_{s=1}^{r-1} \frac{\partial \mathcal{L}_{\text{dyn}}^{t,s}}{\partial b_i}\right),
\end{aligned} \tag{76}$$

where $\mathbf{b}^{t,r} = [b_1^{t,r}, \ldots, b_C^{t,r}]$ denotes the bias parameters in the output layer of the local model at local epoch r, and where $\Delta \mathbf{b}^{\tau} = [\Delta b_1^{\tau}, \ldots, \Delta b_C^{\tau}]$ is the local update of the bias at round τ. Since

$$\frac{\partial \mathcal{L}_{\text{dyn}}^{t,1}}{\partial b_i} = -\mu \sum_{\tau=1}^{t-1} \Delta b_i^{\tau}, \tag{77}$$

it holds that

$$\frac{\partial \mathcal{L}_{\text{dyn}}^{t,2}}{\partial b_i} = -\mu \sum_{\tau=1}^{t-1} \Delta b_i^{\tau} + \mu \eta_t \left(-\mu \sum_{\tau=1}^{t-1} \Delta b_i^{\tau}\right) = -\mu(1+\mu\eta_t) \sum_{\tau=1}^{t-1} \Delta b_i^{\tau} \tag{78}$$

and

$$\frac{\partial \mathcal{L}_{\text{dyn}}^{t,3}}{\partial b_i} = -\mu \sum_{\tau=1}^{t-1} \Delta b_i^{\tau} + \mu \eta_t \left(-\mu \sum_{\tau=1}^{t-1} \Delta b_i^{\tau} - (\mu + \mu^2 \eta_t) \sum_{\tau=1}^{t-1} \Delta b_i^{\tau}\right) = -\mu(1+\mu\eta_t)^2 \sum_{\tau=1}^{t-1} \Delta b_i^{\tau}. \tag{79}$$

By induction,

$$\frac{\partial \mathcal{L}_{\text{dyn}}^{t,r}}{\partial b_i} = -\mu(1+\mu\eta_t)^{r-1} \sum_{\tau=1}^{t-1} \Delta b_i^{\tau}. \tag{80}$$

Therefore, the expectation of the local update of bias in the output layer at round t can be computed as

$$\mathbb{E}\left[\Delta b_i^{(k),t}\right] = \sum_{r=1}^{R} (1+\mu\eta_t)^{r-1} \mu\eta_t \sum_{\tau=1}^{t-1} \mathbb{E}\left[\Delta b_i^{(k),\tau}\right] \tag{81}$$

$$= \left((1+\mu\eta_t)^R - 1\right) \sum_{\tau=1}^{t-1} \mathbb{E}\left[\Delta b_i^{(k),\tau}\right]. \tag{82}$$

Since the objective function of $\mathbb{E}\left[\Delta b_i^{(k),1}\right]$ coincides with that of FedAvg,

$$\mathbb{E}\left[\Delta b_i^{(k),1}\right] = \eta_1 R \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i\right), \tag{83}$$

where η_1 is the learning rate at global round $t = 1$. Then,

$$\mathbb{E}\left[\Delta b_i^{(k),2}\right] = \eta_1 R \left((1+\mu\eta_2)^R - 1\right) \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i\right) \tag{84}$$

$$= a_1 a_2 \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i\right), \tag{85}$$

where $a_1 = \eta_1 R$ and $a_2 = (1+\mu\eta_2)^R - 1$. Furthermore,

$$\mathbb{E}\left[\Delta b_i^{(k),3}\right] = a_1 a_3 (1+a_2)\left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i\right), \tag{86}$$

$$\mathbb{E}\left[\Delta b_i^{(k),4}\right] = a_1 a_4 (1+a_2+a_3+a_2 a_3)\left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i\right), \tag{87}$$

and

$$\mathbb{E}\left[\Delta b_i^{(k),5}\right] = a_1 a_5 (1+a_2+a_3+a_4+a_2 a_3+a_3 a_4+a_2 a_3 a_4)\left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i\right). \tag{88}$$

By induction,

$$\mathbb{E}\left[\Delta b_i^{(k),t}\right] = \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i\right) a_1 a_t \cdot \left(1 + \sum_{i=0}^{t-3}\sum_{\tau=2}^{t-1} \mathbb{I}(\tau+i<t) \prod_{i=\tau}^{\tau+i} a_s\right) \tag{89}$$

$$= a\left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i\right), \tag{90}$$

where $a_t = (1+\mu\eta_t)^R - 1$ and $a = a_1 a_t \left(1 + \sum_{i=0}^{t-3}\sum_{\tau=2}^{t-1} \mathbb{I}(\tau+i<t) \prod_{i=\tau}^{\tau+i} a_s\right) > 0$. After comparing Eq. (89) with its counterpart in the case of FedAvg, we conclude that the previously presented analysis of HiCS-FL extends to FedDyn.

A.8.3 Model-Contrastive Federated Learning (Moon)

Moon [16] relies on the objective function with a contrastive term

$$\mathcal{L}_{\text{moon}} = \frac{1}{Bl}\sum_{j=1}^{l}\sum_{n=1}^{B} \mathcal{L}_{\mathbf{CE}}^{(j,n)} - \mu \log \frac{\exp(\text{sim}(\mathbf{z}^{(j,n)}, \mathbf{z}_{\text{glob}}^{(j,n)})/T)}{\exp(\text{sim}(\mathbf{z}^{(j,n)}, \mathbf{z}_{\text{glob}}^{(j,n)})/T) + \exp(\text{sim}(\mathbf{z}^{(j,n)}, \mathbf{z}_{\text{prev}}^{(j,n)})/T)}, \tag{91}$$

where $\mathbf{z}^{(j,n)}$ denotes the output of the feature extractor of the local model θ_k^t, $\mathbf{z}_{\text{glob}}^{(j,n)}$ is the output of the feature extractor of the global model θ^t, and $\mathbf{z}_{\text{prev}}^{(j,n)}$ is the output of the feature extractor of the local model in the previous round θ_k^{t-1}. Since the contrastive term does not depend on the parameters of bias in the output layer, it holds that

$$\frac{\partial \mathcal{L}_{\text{moon}}^{(j,n)}}{\partial b_i} = \frac{\partial \mathcal{L}_{\mathbf{CE}}^{(j,n)}}{\partial b_i}. \tag{92}$$

Since the expectation of the local updates of bias in the output layer coincides with the one in case of FedAvg, previously presented analysis of HiCS-FL extends to Moon.

A.9 Optimization Algorithms Beyond SGD

Optimizers beyond SGD utilize different model update rules which in principle may lead to different properties of the local update of the bias in the output layer. However, for several variants of SGD, the properties of the local update of the bias remain such that our presented analysis still applies.

A.9.1 SGD with momentum

In each local epoch r, SGD with momentum updates the model according to

$$m_k^{t,r} = \mu m_k^{t,r-1} + (1-\mu)\nabla \mathcal{L}_{\mathbf{CE}}^{t,r}, \tag{93}$$

$$g_k^{t,r} = m_k^{t,r}, \tag{94}$$

$$\theta_k^{t,r} = \theta_k^{t,r-1} - \eta_t g_k^{t,r}, \tag{95}$$

where $m_k^{t,r}$ denotes the momentum in the r-th local epoch, μ is the weight for the momentum, and $m_k^{t,1} = \nabla \mathcal{L}_{\mathbf{CE}}^{t,1}$. Then

$$\Delta\theta_k^t = -\eta_t \sum_{r=1}^{R} g_k^{t,r}, \tag{96}$$

where

$$m_k^{t,1} = \nabla \mathcal{L}_{\mathbf{CE}}^{t,1}, \tag{97}$$

$$m_k^{t,2} = \mu \nabla \mathcal{L}_{\mathbf{CE}}^{t,1} + (1-\mu)\nabla \mathcal{L}_{\mathbf{CE}}^{t,2}, \tag{98}$$

$$\begin{aligned} m_k^{t,3} &= \mu \nabla \mathcal{L}_{\mathbf{CE}}^{t,2} + (1-\mu)\nabla \mathcal{L}_{\mathbf{CE}}^{t,3} \\ &= \mu^2 \nabla \mathcal{L}_{\mathbf{CE}}^{t,1} + \mu(1-\mu)\nabla \mathcal{L}_{\mathbf{CE}}^{t,2} + (1-\mu)\nabla \mathcal{L}_{\mathbf{CE}}^{t,3}. \end{aligned} \tag{99}$$

Therefore,

$$m_k^{t,r} = \mu^{r-1} \nabla \mathcal{L}_{\mathbf{CE}}^{t,1} + (1-\mu) \sum_{\tau=2}^{r} \mu^{r-\tau} \nabla \mathcal{L}_{\mathbf{CE}}^{t,\tau} \tag{100}$$

and thus we have

$$\Delta\theta_k^t = -\eta_t \left(\sum_{r=2}^{R} \left(\mu^{r-1} \nabla \mathcal{L}_{\mathbf{CE}}^{t,1} + (1-\mu) \sum_{\tau=2}^{r} \mu^{r-\tau} \nabla \mathcal{L}_{\mathbf{CE}}^{t,\tau} \right) + \nabla \mathcal{L}_{\mathbf{CE}}^{t,1} \right). \tag{101}$$

Similar to the discussion in the previous section,

$$\mathbb{E}\left[\Delta b_i^{(k)}\right] = \eta_t \left(\sum_{r=2}^{R} \left(\mu^{r-1} + (1-\mu) \sum_{\tau=2}^{r} \mu^{r-\tau} \right) + 1 \right) \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right) \tag{102}$$

$$= a \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right) \tag{103}$$

where $a = \eta_t \left(\sum_{r=2}^{R} \left(\mu^{r-1} + (1-\mu) \sum_{\tau=2}^{r} \mu^{r-\tau} \right) + 1 \right) > 0$. Similar result is obtained when SGD applies Nesterov acceleration as long as the optimizers are not using second-order momentum.

A.9.2 Adam Optimizer

Recall that the two observations regarding the gradient of $\mathcal{L}_{\mathbf{CE}}$ still hold when training the model with an adaptive optimizer such as Adam [15]. However, Adam updates the model differently from SGD. In particular, each entry of the gradient has an adaptive learning rate tied to its magnitude. With an SGD optimizer, the magnitude of the i-th entry of the local update of bias $\Delta\mathbf{b}^{(k)}$ is approximately proportional to the fraction of the samples with label i, $D_i^{(k)}$ (if $\mathcal{E}_i$ is small),

$$\mathbb{E}\left[\Delta b_i^{(k)}\right] = \eta_t R \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right). \tag{104}$$

However, this observation does not hold when using the Adam optimizer for the local update because each entry has a different learning rate $\eta_{t,i}$ and thus

$$\mathbb{E}\left[\Delta b_i^{(k)}\right] = \eta_{t,i} R \left(D_i^{(k)} \sum_{c=1}^{C} \mathcal{E}_c - \mathcal{E}_i \right). \tag{105}$$

Although the magnitude of $\mathbb{E}\left[\Delta b_i^{(k)}\right]$ is no longer approximately proportional to $D_i^{(k)}$, we can utilize the sign of $\mathbb{E}\left[\Delta b_i^{(k)}\right]$, i.e.,

$$\text{if } D_i^{(k)} \gg D_j^{(k)}, \text{ then } \mathbb{P}\left(\mathbb{E}\left[\Delta b_i^{(k)}\right] > 0\right) \gg \mathbb{P}\left(\mathbb{E}\left[\Delta b_j^{(k)}\right] > 0\right). \tag{106}$$

Suppose client k has highly imbalanced data, i.e., $H(\mathcal{D}^{(k)})$ is small. Then the maximal component $\max_i D_i^{(k)}$ is much larger than the other components; in fact, it is likely to have only one positive

component in the local update of bias $\Delta \mathbf{b}^{(k)}$. On the contrary, suppose client u has balanced data and thus $H(\mathcal{D}^{(u)})$ is large. The maximal component $\max_i D_i^{(u)}$ is then very close to the other components, and it is likely to observe larger number of positive components in the local update of $\Delta \mathbf{b}^{(u)}$. While characterizing $\mathbb{P}(\mathbb{E}[\Delta b_i^{(k)}] > 0)$ appears challenging, we can empirically infer that client u with more balanced data has a local update of bias $\Delta \mathbf{b}^{(u)}$ with more positive components. With

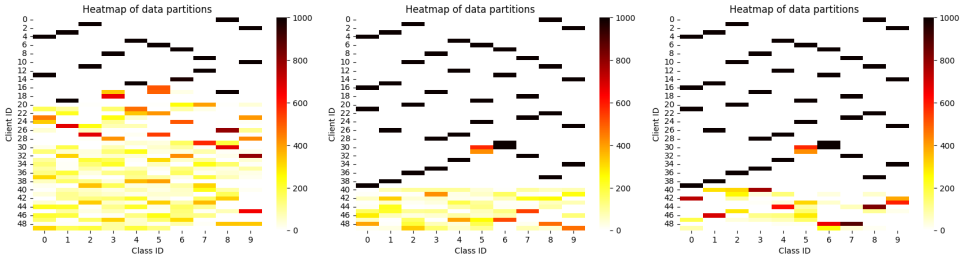
$$\hat{H}(\mathcal{D}^{(u)}) \triangleq H(\text{softmax}(\Delta \mathbf{b}^{(u)}, T)), \tag{107}$$

$$\hat{H}(\mathcal{D}^{(k)}) \triangleq H(\text{softmax}(\Delta \mathbf{b}^{(k)}, T)), \tag{108}$$

$\hat{H}(\mathcal{D}^{(u)})$ is more likely to be larger than $\hat{H}(\mathcal{D}^{(k)})$. The examples of estimated entropy when utilizing Adam as the optimizer are provided in Section. A.12.

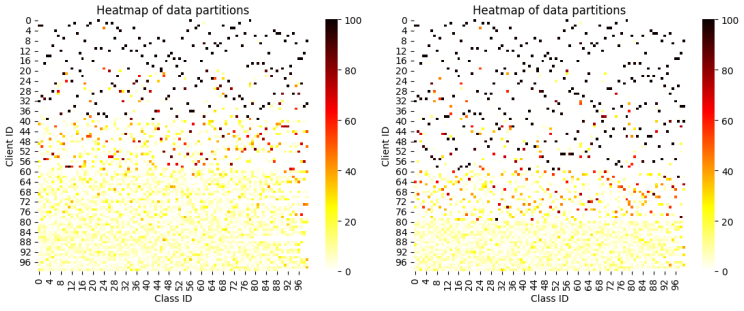
A.10 Visualization of Data Partitions

To generate non-IID data partitions we follow the strategy in [35], utilizing Dirichlet distribution with different concentration parameters α to control the heterogeneity levels. In particular, the number of samples with label i owned by client k is set to $\frac{X_i^{(k)} N_i}{\sum_{j=1}^{N} X_i^{(j)}}$,where $X_i^{(1)}, \ldots, X_i^{(N)}$ are drawn from $\text{Dir}(\alpha)$ and N_i denotes the total number of samples with label i in the overall dataset. For the setting with multiple α, we divide the overall training set into $|\alpha|$ equal parts and generate data partitions according to the method above. Figures 6 and 7 illustrate the class distribution of local clients by displaying the number of samples with different labels; colors distinguish between magnitudes – the darker the color, the more samples are in the class.



(a) (b) (c)

Figure 6: Results on CIFAR10. Training data is split into 50 partitions according to a Dirichlet distribution (50 clients). The concentration parameter is as follows: (1) $\alpha \in \{0.001, 0.01, 0.1, 0.5, 1.0\}$; (2) $\alpha \in \{0.001, 0.002, 0.005, 0.01, 0.5\}$; (3) $\alpha \in \{0.001, 0.002, 0.005, 0.01, 0.1\}$. The figures (a), (b) and (c) correspond to settings (1), (2) and (3), respectively.



(a) (b)

Figure 7: Results on Mini-ImageNet. Training data is split into 100 partitions according to Dirichlet distribution (100 clients). The concentration parameter is varied as follows: (1) $\alpha \in \{0.001, 0.01, 0.1, 0.5, 1.0\}$; (2) $\alpha \in \{0.001, 0.005, 0.01, 0.1, 1.0\}$. The figures (a) and (b) correspond to settings (1) and (2), respectively.

Table 5: The columns "Extra Computation" and "Extra Communication" denote the computation and communication complexity of additional operations in each sampling scheme compared to random sampling.

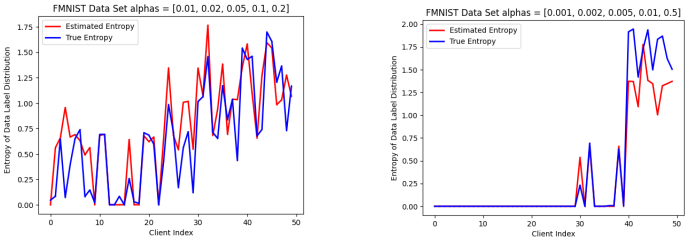
Method	Extra Computation	Extra Communication
Random	-	-
pow-d	$\mathcal{O}(\lvert\theta^t\rvert)$	$\mathcal{O}(\lvert\theta^t\rvert)$
CS	$\mathcal{O}(\lvert\theta^t\rvert)$	-
DivFL	$\mathcal{O}(\lvert\theta^t\rvert)$	$\mathcal{O}(\lvert\theta^t\rvert)$
FedCor	$\mathcal{O}(\lvert\theta^t\rvert)$	-
HiCS-FL	$\mathcal{O}(C)$	-

A.11 Computational and Communication Complexity

We compare the communication and computational costs of HiCS-FL with those of the competing methods, including Power of Choice (pow-d) [8], Clustered Sampling [11] and DivFL [2], and map them against random sampling, as shown in Table. 5. In its ideal setting, pow-d selects K clients with the largest local validation loss among all N clients. To compute the local validation loss at the beginning of a global training round t, the server must send the global model to all clients. Compared to the random sampling strategy where the global model is sent to only K clients, pow-d must transmit additional $(N-K)|\theta^t|$ model parameters. Moreover, pow-d requires all clients to compute validation loss of the global model θ^t on local datasets, which incurs additional $\mathcal{O}(N|\theta^t|)$ computations. While communication requirements of Clustered Sampling do not exceed those of random sampling, the server must run a clustering algorithm on the local updates of dimension $|\theta^t|$ (the same as gradients). DivFL relies on maximizing a submodular function to select the most diverse clients based on all clients' gradients, leading to a transmission overhead and additional computation involving $|\theta^t|$-dimensional gradients. In our experiments, DivFL has consistently required the longest training time and memory usage due to its dependence on the submodularity maximizer. FedCor [28] cliams that only partial clients participating in the global update after warm-up stage but still needs all clients to perform inference for computing validation loss in the warm-up stage. Our proposed method, HiCS-FL, does not require any additional transmission of model parameters; furthermore, in HiCS-FL the server clusters clients based on their local updates of the bias in the output layer, which is low-dimensional and model-agnostic. Overall, HiCS-FL requires negligible computational overhead to significantly improve the performance of non-iid Federated Learning.

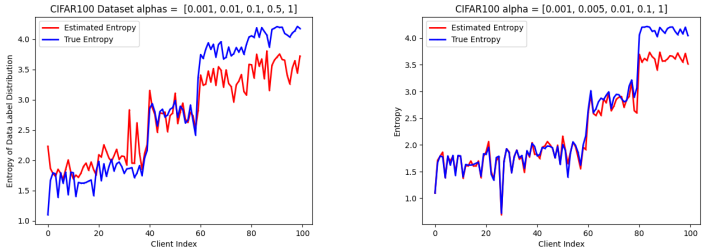
A.12 Examples of Estimated Entropy

To further illustrate the proposed framework, here we show a comparison between the estimated entropy of data label distribution and the true entropy. Specifically, Figures 8 and 9 show that the entropy estimated by the proposed method is close to the true entropy; the experiments were conducted on FMNIST and Mini-ImageNet, using SGD and Adam as optimizers, respectively. As stated in Theorem 3.3, the clients with larger true entropy are likely to have lager estimated entropy. In case where the model is trained with Adam, estimated entropy of data label distribution is not as accurate as in the case of using SGD. Figures 10 and 11 compare the performance of estimating entropy with SGD and Adam optimizers for the same setting of α. Notably, as shown in the figures, the method is capable of distinguishing clients with extremely imbalanced data from those with balanced data.



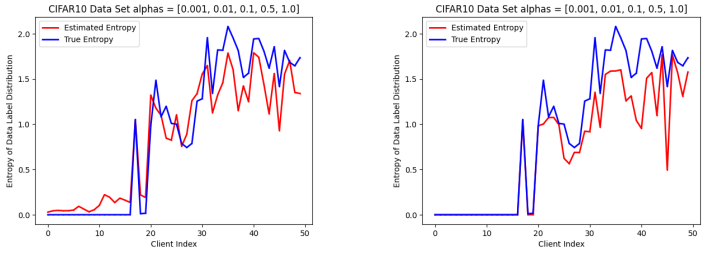
(a) (b)

Figure 8: The estimated entropy of data label distribution in experiments on FMNIST with SGD as the optimizer. The parameter α for the two figures: (a) $\alpha \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$; (b) $\alpha \in \{0.001, 0.002, 0.005, 0.01, 0.5\}$



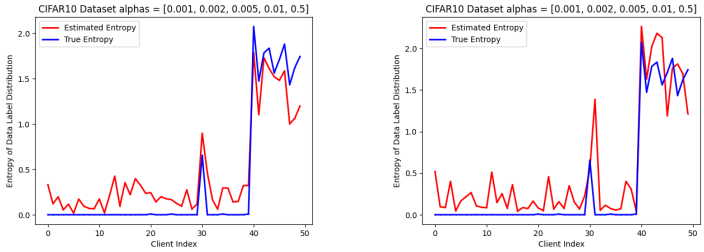
(a) (b)

Figure 9: The estimated entropy of data label distribution in experiments on Mini-ImageNet with Adam as the optimizer. The parameter α for the two figures: (a) $\alpha \in \{0.001, 0.01, 0.1, 0.5, 1.0\}$; (b) $\alpha \in \{0.001, 0.005, 0.01, 0.1, 1.0\}$.



(a) (b)

Figure 10: The estimated entropy of data label distribution in experiments on CIFAR10 with $\alpha \in \{0.001, 0.01, 0.1, 0.5, 1.0\}$. (a) The result of the experiments using SGD as the optimizer. (b) The result of the experiments using Adam as the optimizer.



(a) (b)

Figure 11: The estimated entropy of data label distribution in experiments on CIFAR10 with $\alpha \in \{0.001, 0.002, 0.005, 0.01, 0.5\}$. (a) The result of the experiments using SGD as the optimizer. (b) The result of the experiments using Adam as the optimizer.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: [NA]

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of exNoecution) needed to reproduce the experiments?

   Answer: [No]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.