Exploiting Label Skews in Federated Learning with Model Concatenation

Yiqun Diao[1], **Qinbin Li**[2], **Bingsheng He**[1]

[1]National University of Singapore
[2]University of California, Berkeley
{yiqun, hebs}@comp.nus.edu.sg, qinbin@berkeley.edu

Abstract

Federated Learning (FL) has emerged as a promising solution to perform deep learning on different data owners without exchanging raw data. However, non-IID data has been a key challenge in FL, which could significantly degrade the accuracy of the final model. Among different non-IID types, label skews have been challenging and common in image classification and other tasks. Instead of averaging the local models in most previous studies, we propose FedConcat, a simple and effective approach that concatenates these local models as the base of the global model to effectively aggregate the local knowledge. To reduce the size of the global model, we adopt the clustering technique to group the clients by their label distributions and collaboratively train a model inside each cluster. We theoretically analyze the advantage of concatenation over averaging by analyzing the information bottleneck of deep neural networks. Experimental results demonstrate that FedConcat achieves significantly higher accuracy than previous state-of-the-art FL methods in various heterogeneous label skew distribution settings and meanwhile has lower communication costs. Our code is publicly available at https://github.com/sjtudyq/FedConcat.

Introduction

A good machine learning model usually needs a large high-quality dataset to train. However, due to privacy concern and regulations such as GDPR (Voigt and Von dem Bussche 2017), sometimes it is not allowed to collect original data for centralized training. Federated learning (FL) (Kairouz et al. 2019; Li et al. 2019a,b; Yang et al. 2019) is proposed to let data owners collaboratively train a better machine learning model without exposing raw data. It has become a hot research topic (Li, Wen, and He 2020; Dai, Low, and Jaillet 2020; He, Annavaram, and Avestimehr 2020; Li et al. 2020a; Karimireddy et al. 2020; Liu et al. 2020; Wu et al. 2020). FL has many potential practical applications (Bonawitz et al. 2019; Hard et al. 2018; Kaissis et al. 2020). For example, different hospitals can collectively train a FL model for diagnosing diseases through medical imaging, while protecting the privacy of individual patients.

A typical framework of FL is FedAvg (McMahan et al. 2016), where the clients train and send their local models to the server, and the server averages the local models to update the global model in each round. It has been shown that data heterogeneity is a challenging problem in FL, since non-IID data distributions among FL clients can degrade the FL model performance and slow down model convergence (Karimireddy et al. 2020; Li et al. 2020b; Hsu, Qi, and Brown 2019; Li et al. 2021). According to Li et al. (2021), non-IID data includes label skews, feature skews and quantity skews. In this paper, we focus on label skews (i.e., the label distributions of different clients are different), which is popular in reality (e.g., disease distributions vary across different areas).

Researchers have put some promising effort to address the above label skew challenge. For example, FedProx (Li et al. 2020a) uses the L_2 distance between the local model and the global model to regularize the local training. MOON (Li, He, and Song 2021) regularizes the local training using the similarity between representations of the local model and the global model. FedRS (Li and Zhan 2021) restricts the updates of unseen classes during local training. FedLC (Zhang et al. 2022) further calibrates logits to reduce the updates of minority classes. The key idea of existing studies is usually to reduce the drift produced in local training (Li et al. 2020a; Li, He, and Song 2021; Karimireddy et al. 2020; Li and Zhan 2021; Zhang et al. 2022) or design a better federated averaging scheme in the server (Wang et al. 2020b,a). Those algorithms are based on the averaging framework. They attempt to address the label skew problem by mitigating its side effect in federated averaging. However, existing methods cannot achieve satisfactory performance. In label skews, the averaging methods may not make much sense, as each party may have very different models to predict different classes. Especially under extreme label skews where each client has quite different classes (e.g., face recognition), since the local optima are far from each other, averaging these local models leads to significant accuracy degradation. Even worse, it is challenging to quantify how label skews influence the model due to the diversity of label skews in practice.

In this paper, we think out of the model-averaging scheme, and propose to use model concatenation as the aggregation method. Since each local model is good at classifying samples of several classes due to label skews, we propose to concatenate the features learned by the local models to combine the knowledge from the local models. For

example, in the label skew setting, one client has sufficient data on cats with little data on dogs, while another client has sufficient data on dogs with little data on cats. Then, each client can train a local model which is good at predicting one class. Intuitively, concatenating those models can gather all key information, which can help train a good classifier for all classes among clients. This seemingly simple idea fundamentally changes the way of existing methods regarding label skews as an issue to avoid or mitigate.

With this idea, we propose a novel FL algorithm to address label skews named FedConcat. First, the server divides clients into a few different clusters according to their label distributions. To address the privacy concern of uploading label distribution information, we develop an effective method to infer label distribution directly from the model. Second, FedAvg is conducted among each cluster to learn a good model for each kind of label distribution. Third, the server concatenates encoders of models of all clusters (i.e. neural networks except the last layer). Finally, with the parameters of the concatenated encoders fixed, the server and the clients jointly train a classifier on top of it using FedAvg. We theoretically justify that concatenation keeps richer mutual information than averaging in the feature space by applying the information bottleneck theory.

Among each cluster, clients have similar label distributions. The label skew problem is alleviated inside the cluster, so FedAvg is competent to train a good model for each cluster with slight label skews. Since the concatenated encoders have already extracted good features, the task of training a linear classifier in the final stage becomes simpler. Therefore, FedAvg can achieve good accuracy for the simplified task. Moreover, through clustering, we can control the size of global model by adjusting the number of clusters.

We conduct extensive experiments with various label skew settings. Our experimental results show that FedConcat can significantly improve the accuracy compared with the other state-of-the-art FL algorithms including FedAvg (McMahan et al. 2016), FedProx (Li et al. 2020a), MOON (Li, He, and Song 2021), FedRS (Li and Zhan 2021) and FedLC (Zhang et al. 2022). The improvement is more significant under extreme label skews. Besides, FedConcat can achieve better accuracy with much smaller communication and computation costs compared with baselines.

Our contributions can be summarized as follow:

- Instead of averaging, we propose a new aggregation method in FL by concatenating the local models. Moreover, we apply clustering technique to alleviate label skew and control the size of global model.
- We theoretically show that concatenation preserves more information than averaging from the information bottleneck perspective, which guarantees the effectiveness of our approach.
- We conduct extensive experiments to show the effectiveness and communication efficiency of FedConcat. Under various label skews of a popular FL benchmark (Li et al. 2021), FedConcat can outperform baselines averagely by 4% on CIFAR-10, by 8% on CIFAR-100, by 2% on Tiny-ImageNet, and by 1% on FMNIST and SVHN datasets.

Background and Related Work

Denote $D^i = (X^i, Y^i)$ the local dataset of client i. Label skews mean that $P(Y^i)$ differs among clients. According to Li et al. (2021), label skews can lead to significant accuracy degradation of the global model. It is also prevalent in real-world scenarios. For example, the disease distributions differ in different regions, which leads to label skews when training a global automatic disease diagnosis system.

Previous studies like FedAvg (McMahan et al. 2016) average all models submitted by clients. However, under the non-IID data distribution cases, each client trains a good local model towards its local optimum. While the local optima may be far from each other, simply averaging the local models may produce a global model that is also far from the global optimum. There are many existing studies aiming to solve the non-IID data distribution problem based on FedAvg (McMahan et al. 2016).

A popular way is to improve local training so that the local model is not too far from the global optimum. For example, FedProx (Li et al. 2020a) adds a regularization term which measures the distance between the local model and the global model. MOON (Li, He, and Song 2021) shares a similar motivation, regularizing by a contrastive loss to measure the distance between representations of the local model and the global model. Both methods add one more term to the loss function and require extra computations than FedAvg. SCAFFOLD (Karimireddy et al. 2020) adjusts the local gradient by keeping a correction term for each client, therefore its communication cost doubles. Wang et al. (2021) propose to monitor the class imbalance of each client based on uploaded gradient together with a small public dataset. Then they mitigate the imbalance by their Ratio Loss. FedRS (Li and Zhan 2021) proposes to restrict the updates of missing classes by down-scaling their logits, however it only deals with missing classes. To further deal with minority classes, FedLC (Zhang et al. 2022) proposes to calibrate logits based on the label statistics of local training data. FedOV (Diao, Li, and He 2023b) introduces the "unknown" class and trains open-set classifiers in local training for a better ensemble. More techniques are discussed in Appendix A.1 and A.2 of our full version (Diao, Li, and He 2023a).

Our Method: FedConcat

Problem Statement

Federated learning aims to train a global model on multiple clients without exposing their raw data. Denote D^i the local dataset of client i. Suppose there are K clients, and the local loss function for each client is $\mathcal{L}(\cdot, \cdot)$. Formally, our goal is to train a global model f that minimizes the following objective.

$$L = \frac{\sum_{i=1}^{K} |D^i| \cdot \mathbb{E}_{(X^i, Y^i) \sim D^i}[\mathcal{L}(f(X^i), Y^i)]}{\sum_{i=1}^{K} |D^i|} \quad (1)$$

Motivation

Pitfalls of existing methods in label skews Under label skews, the local models can be much different as they are

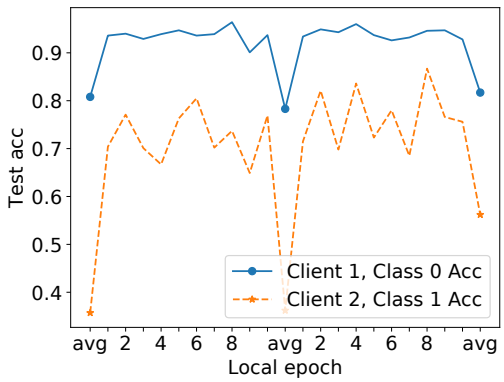

Figure 1: Accuracy of local models and averaged model on two clients under label skews.

trained on different classes. Therefore it hardly makes sense to average each parameter of these models with quite different tasks. As an example, we train FedAvg on two clients of CIFAR-10 under label skews. The first client only has samples of class 0 and 2, while the second client only has samples of class 1 and 9. For both clients, we train 10 local epochs per round. We show the accuracy of local models and averaged global model of two rounds in Figure 1. As we can see, the accuracy of local classes increases during local training, while the averaging operation leads to significant accuracy degradation. This example illustrates the problem of averaging local models under extreme label skews.

An alternative view of label skews Let us view the neural network as a feature extractor (all the layers in the network except the last layer) and a classifier (the last layer). Since each client's model is well-fitted in its own dataset, we already have quite a few locally well-trained feature extractors. Intuitively, concatenating the features from different local extractors can provide a better feature representation for label skews. Thus we propose the idea of concatenating feature extractors and training a global classifier.

If we concatenate the models of all clients, our final model size can grow much large if there are many clients, and the overhead of training the global classifier is much more expensive. In practice, although label skews are prevalent, some parties may have similar label distributions. For example, hospitals in the same region may encounter similar types of diseases. Therefore, we adopt the clustering method before training. By clustering all clients into a few groups via their label distributions, we can control the size of global model. Inside each group, since grouped clients have similar label distributions, the trained model can capture this kind of data well.

In brief, we tackle the label skew problem by generating solutions for each group individually. Next, we combine those solutions together to get a better global model with smaller communication cost.

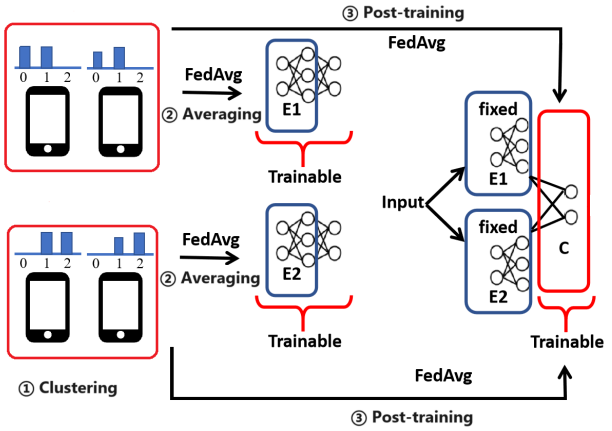

Figure 2: The workflow of FedConcat. (1) Clustering stage: clients are clustered based on label distributions; (2) Averaging stage: each cluster trains a model using FedAvg; (3) Post-training stage: all well-trained feature extractors (E1, E2) are concatenated. All clients train a global classifier (C) collectively with feature extractors fixed. For FedConcat-ID, label distributions are inferred in the clustering stage.

Proposed Algorithm

Our framework is illustrated in Figure 2. It has three stages: clustering, averaging and post-training. First, clients with similar label distributions are grouped into same cluster. Then, each cluster performs FL to train a model that fits well inside the cluster. Finally, the server collects feature extractors of all clusters with their parameters fixed, and train a global classifier among all clients. The overall algorithm is shown in Algorithm 1. In the following, we elaborate those stages in detail.

Stage 1-A: Clustering with label distributions In order to alleviate the label imbalance problem, we perform clustering based on label distributions, so that each cluster hosts clients with similar label distributions. Formally, for client i, suppose there are $N_{i,j}$ samples of class j, and there are a total of $N_i = \sum_j N_{i,j}$ samples. Its label distribution is defined as vector

$$P_i(y) = (\frac{N_{i,1}}{N_i}, \frac{N_{i,2}}{N_i}, ..., \frac{N_{i,m}}{N_i}), \tag{2}$$

where there are m classes globally. In this paper, we use K-means algorithm (Lloyd 1982) to perform clustering. For the hyper-parameter K, one can utilize elbow method to select the best value. We use K-means as it is simple, popular and sufficiently good for our study. With clustering, we can control the number of different models generated by the clients, which helps to reduce the model size in our later concatenation.

Stage 1-B: Clustering without label distributions If clients are unable to upload label distributions due to privacy concerns, we propose to utilize the uploaded local models of the first round to infer the approximate label distribution of each client. In this way, we only upload trained models like FedAvg, which does not cause any extra privacy leakage.

During local training, if a class appears more frequently, the model is prone to output a higher probability for that class. Many works (Johnson and Khoshgoftaar 2019; Bahng et al. 2020) have observed that predictions of deep learning model are biased towards the majority classes of the training set. Intuitively, if we put a large batch of random inputs into the client model, the average prediction can indicate the label distribution of training data. Thus, we generate random data (i.e., images that each pixel is randomly generated from range zero to one) and input these random data into each client model. Then, we calculate the average prediction probability for each class as the inferred label distribution of each client. Formally, denote the model of client i as f_i. We randomly generate r inputs $X_1, ..., X_r$, the inferred distribution of client i

$$P_i^{ID}(y) = \frac{1}{r}\sum_{j=1}^{r}\sigma(f_i(X_j)), \tag{3}$$

where σ is the softmax function.

We refer to this variant as FedConcat with Inferred Distribution (FedConcat-ID). A neural network classifier can be viewed as a function $p(Y|X)$ learned on its training data. In an ideal scenario, if the inputs X are independent of Y, the equation $p(Y) = p(Y|X)$ holds true. The underlying intuition of Eq. (3) is to employ uninformative inputs to approximate $p(Y)$.

Stage 2: Averaging Within each cluster, we use FedAvg (McMahan et al. 2016) to train a model that fits well for such cluster. Inside a cluster, since the label distributions of the clients are similar, we expect the global model to have a good performance on the dominant classes of the cluster.

Stage 3: Post-training Now that we have K models, we stack their encoders (all layers but the last layer) as the global feature extractor. Then we broadcast the global feature extractor to all clients for one time, and ask clients to jointly train a classifier using FedAvg, with the global feature extractor fixed. Since the encoder training is stopped, we can calculate the features of raw data in a forward pass for only one time. For other training rounds, we can directly feed features into the linear classifier to train it. Therefore in this stage, our major computation and communication happens only for the linear classifier.

Algorithm 1: FedConcat and FedConcat-ID

Input: number of clients N, number of clusters K, number of training rounds of the encoder T_e, number of training rounds of the classifier T_c
Output: the final model w

if FedConcat **then**
  $S_1, S_2, ..., S_K \leftarrow Kmeans(P_i(y)_{i=1}^N)$ // Perform K-means based on label distributions
if FedConcat-ID **then**
  Initialize global model f_g
  for $i = 1, 2, ..., N$ in parallel **do**
    $f_i \leftarrow TrainLocal(f_g)$ // Send model to each client for local training
  $S_1, S_2, ..., S_K \leftarrow Kmeans(P_i^{ID}(y)_{i=1}^N)$ // Infer label distributions by Eq. (3) and perform K-means
Initialize encoder E_i and classifier C_i for each cluster
for $t = 1, 2, ..., T_e$ **do**
  for $i = 1, 2, ..., K$ **do**
    $E_i, C_i \leftarrow FedAvg(\{E_i, C_i\}, S_i)$ // Run FedAvg to train encoder and classifier for each cluster
$E = \{E_1, E_2, ..., E_K\}$
Initialize global classifier C
for $t = 1, 2, ..., T_c$ **do**
  $C \leftarrow FedAvg(C, \bigcup_{i=1}^K S_i)$ // Fix E and run FedAvg on all clients to train C
return final model $w = \{E, C\}$

Theoretical Analysis and Discussion

Analyzing FedConcat by Information Bottleneck

To answer the question why concatenating encoders works for extreme label skews, we refer to the information bottleneck theory (Shwartz-Ziv and Tishby 2017). Suppose a deep neural network f is trained on dataset D. Denote a random train sample $(X, Y) \sim D$ where X are input variables and Y are desired outputs. Suppose the extracted features (representation before the last fully-connected layer) is Z, the neural network learns an encoder which minimizes

$$\mathbf{E}_{(X,Y)\sim D}[I(X;Z) - \beta I(Z;Y)], \tag{4}$$

where $I(\cdot;\cdot)$ denotes the mutual information between two variables and β is a positive trade-off parameter related to the task.

In brief, the encoder of deep neural network aims to remember the features related to the target outputs (maximizing $I(Z;Y)$), while forgetting the information of inputs unrelated to the target outputs (minimizing $I(X;Z)$).

Consider the label skews in federated learning. Suppose there are two clients with local datasets D^1 and D^2 respectively. Their locally trained encoders are f_{e1} and f_{e2}. Denote the FedConcat encoder as $f_e(\cdot) = \{f_{e1}(\cdot), f_{e2}(\cdot)\}$ and the FedAvg encoder as f_{avg}. We have the following theorem.

Theorem 1. *$I(f_{avg}(X);Y) < I(f_e(X);Y)$, $\forall(X,Y) \sim D^1 \cup D^2$.*

Proof. According to the information bottleneck theory, the local model of the first client minimizes

$$\mathbf{E}_{(X^1,Y^1)\sim D^1}[I(X^1; f_{e1}(X^1)) - \beta_1 I(f_{e1}(X^1); Y^1)]. \tag{5}$$

Similarly, the second client's local model minimizes

$$\mathbf{E}_{(X^2,Y^2)\sim D^2}[I(X^2; f_{e2}(X^2)) - \beta_2 I(f_{e2}(X^2); Y^2)]. \tag{6}$$

For a good global encoder f_e, it should minimize

$$\mathbf{E}_{(X,Y)\sim D^1\cup D^2}[I(X; f_e(X)) - \beta I(f_e(X); Y)]. \tag{7}$$

For the mutual information between representation and target, no matter whether $(X,Y) \sim D^1$ or $(X,Y) \sim D^2$, we have

$$I(f_e(X);Y) \geq \max\{I(f_{e1}(X);Y), I(f_{e2}(X);Y)\}, \tag{8}$$

which means the representation of concatenated encoders are more related to the global targets than single locally optimized encoder.

For the part of forgetting task-unrelated information, we have

$$I(f_e(X);X) \geq \max\{I(f_{e1}(X);X), I(f_{e2}(X);X)\}, \quad (9)$$

which is a disadvantage according to information bottleneck theory. According to experimental results in Shwartz-Ziv and Tishby (2017), when deep neural network reaches convergence, the mutual information between last layer representation and raw input (i.e. $I(f_e(X);X)$) becomes very small, as compared to $I(f_e(X);Y)$. Therefore, we regard $I(f_e(X);Y)$ as the main part. We also justify such hypothesis by experiments, which are included in Appendix C of our full version (Diao, Li, and He 2023a).

For the averaging solution, under label skews, the averaged global model becomes very different from local optima. Thus, for $(X,Y) \sim D^1$, we have

$$I(f_{avg}(X);Y) < I(f_{e1}(X);Y) \leq I(f_e(X);Y). \quad (10)$$

Similarly, for $(X,Y) \sim D^2$, we have

$$I(f_{avg}(X);Y) < I(f_{e2}(X);Y) \leq I(f_e(X);Y). \quad (11)$$

Combining Eq. (10) and (11), for $(X,Y) \sim D^1 \cup D^2$, we have $I(f_{avg}(X);Y) < I(f_e(X);Y)$.

□

Implication Under label skews, the averaged encoder contains less mutual information about the labels compared with the concatenation of well-trained local encoders. This explains why the averaged model suffers from accuracy decay compared with local models, as shown in Figure 1.

Privacy

FedConcat needs client label distribution information, therefore it is applicable when local label distribution is not sensitive. For scenarios that also consider label distribution privacy, users can adopt FedConcat-ID, which only transfers the models and provides the same privacy level as FedAvg. The inference attack towards client model is a complex topic and defense mechanisms against them fall outside this paper's scope. It is an interesting topic to explore more robust measures to prevent such breaches in future works.

Communication

Suppose the model size is w, and its last classifier layer size is cw ($c < 1$). For T_e encoder rounds, each client's communication cost is $2T_e w$. Downloading the concatenated model costs Kw. Next, for classifier rounds each client costs $2T_c Kcw$. The total cost of FedConcat is $2wN(T_e + K/2 + cKT_c)$. For FedAvg with T rounds, the cost is $2wNT$. Suppose $T = T_e + T_c$ (i.e., we train the same communication rounds for FedAvg and FedConcat). Given the same model size w and the number of clients N, communication overhead can be saved by choosing small c, K, i.e. limiting the classifier size and number of cluster. Experimental results in Appendix D.2 of our full version (Diao, Li, and He 2023a) verify that our approach achieves higher accuracy and more stable convergence given the same communication cost with FedAvg and other baselines.

Experiments

We have conducted extensive experiments to evaluate our method. Through comprehensive experiments, we find that our techniques consistently outperform baseline methods, delivering superior accuracy and more stable convergence under various label skews. Importantly, our methods remain effective in scenarios characterized by partial client participation, large models, and an increased number of clients. The introduced label inference and clustering components are both straightforward and effective. Due to the space limit, we put the following experiments in Appendix of our full version (Diao, Li, and He 2023a).

- D.2: training curves with communication costs.
- D.3: training curves with computation costs.
- D.4: compared to baselines on the concatenated model.
- D.5: varying the number of clusters.
- D.6: more results on varying the clustering strategies.
- D.7: client partial participation settings.
- D.8: more results and analysis on large models.
- D.9: more results on varying the number of clients.
- D.10: analyzing FedConcat-ID label inference module.

Experiment Setups

Datasets Our experiments engage CIFAR-10 (Krizhevsky, Hinton et al. 2009), FMNIST (Xiao, Rasul, and Vollgraf 2017), SVHN (Netzer et al. 2011), CIFAR-100 (Krizhevsky, Hinton et al. 2009), and Tiny-ImageNet datasets (Wu, Zhang, and Xu 2017) to evaluate our algorithm. The partition strategy from Li et al. (2021) generates various non-IID settings, with a focus on label skews, given their significant accuracy degradation (Li et al. 2021). In experiments, $\#C = k$ represents clients with k unique labels, while $p_k \sim Dir(\beta)$ denotes the Dirichlet distribution sampled proportion of each class samples assigned to each client. By default, we divide whole dataset into 40 clients.

Baselines Our method is compared with well-known, open-sourced FL methods including FedAvg (McMahan et al. 2016), FedProx (Li et al. 2020a), MOON (Li, He, and Song 2021), FedRS (Li and Zhan 2021), and FedLC (Zhang et al. 2022). The baseline settings replicate those from Li et al. (2021), running 50 rounds with each client training 10 local epochs per round, batch size 64, and learning rate 0.01 using SGD optimizer with weight decay 10^{-5}.

Models To investigate diverse scenarios with different clients' capacities, we experiment with three different neural networks: simple CNN, VGG-9, and ResNet-50. By default, we use simple CNN. Appendix D.1 of our full version (Diao, Li, and He 2023a) provides more details on the setups.

Effectiveness

We evaluate the performance of FedConcat and FedConcat-ID against other baselines. By default, our configuration includes a division of the 40 clients into $K = 5$ clusters, and 200 rounds allocated for training the classifier. In order to equate the communication cost of FedConcat to that of 50

Dataset	Partition	FedAvg	FedProx	MOON	FedRS	FedLC	FedConcat	FedConcat-ID
CIFAR-10	$\#C = 2$	53.6%	53.1%	53.4%	53.8%	49.8%	**56.9%**	**56.5%**
	$\#C = 3$	57.6%	57.4%	58.6%	59.1%	58.1%	**62.0%**	**61.8%**
	$p_k \sim Dir(0.1)$	53.0%	52.8%	53.1%	54.8%	53.7%	**57.7%**	**56.9%**
	$p_k \sim Dir(0.5)$	59.9%	59.9%	61.2%	61.5%	61.5%	**64.2%**	**63.7%**
SVHN	$\#C = 2$	82.8%	82.6%	83.0%	79.5%	75.7%	**83.4%**	**83.2%**
	$\#C = 3$	85.2%	85.2%	84.7%	85.7%	84.8%	**86.0%**	**86.1%**
	$p_k \sim Dir(0.1)$	**84.0%**	83.9%	83.7%	80.9%	78.8%	83.2%	82.9%
	$p_k \sim Dir(0.5)$	87.2%	87.2%	87.2%	87.1%	87.1%	**87.5%**	**87.9%**
FMNIST	$\#C = 2$	79.0%	81.8%	81.4%	78.3%	77.7%	**84.4%**	**83.0%**
	$\#C = 3$	84.7%	85.7%	84.6%	85.8%	86.0%	**87.1%**	**86.6%**
	$p_k \sim Dir(0.1)$	85.1%	**85.2%**	85.0%	82.5%	82.1%	84.5%	85.0%
	$p_k \sim Dir(0.5)$	87.5%	87.4%	87.4%	87.5%	87.5%	**87.7%**	**87.5%**

Table 1: Experimental results of our methods compared with baselines with same communication cost. The model of baseline algorithms is the model of one cluster in FedConcat. We run three different random seeds and report the average accuracy.

Partition	Best baseline	FedConcat	FedConcat-ID
$\#C = 2$	48.6%	**48.9%**	44.6%
$\#C = 3$	51.5%	**53.1%**	**51.8%**
$p_k \sim Dir(0.1)$	43.7%	**47.6%**	**46.7%**
$p_k \sim Dir(0.5)$	54.4%	**56.7%**	**56.8%**

Table 2: Scalability of FedConcat and FedConcat-ID compared with baselines on CIFAR-10, 200 clients.

rounds of FedAvg, we set the encoder training to 31 rounds. For the FMNIST dataset, due to its image size differing from CIFAR-10 and SVHN, we record the test accuracy at classifier round 173 to maintain similar communication costs.

Results in Table 1 illustrate that FedConcat consistently outperforms the other five FL algorithms in most scenarios. Specifically, in the challenging CIFAR-10 dataset, both FedConcat and FedConcat-ID offer an average improvement of about 4%. When considering partition types, notable improvements are evident in the more complex $\#C = 2$ and $\#C = 3$ partitions. For the Dirichlet-based label distributions of SVHN and FMNIST datasets, since the label skews are slight, the accuracy degradation of baseline algorithms from centralized training is small. In such scenarios, our methods exhibit comparable accuracy with the baselines.

Scalability

In this section, we evaluate the scalability of FedConcat. We keep the number of clusters $K = 5$. During each round, a random selection of 50% of the clients is sampled to participate in FL training. The results, as illustrated in Table 2, confirm that both FedConcat and FedConcat-ID continue to outperform baseline algorithms with 200 clients and partial participation settings.

Experiments on Larger Model

In this section, we conduct experiments on larger models, more clients and more complicated tasks, i.e. training ResNet-50 on CIFAR-100 and Tiny-ImageNet. There are 200 clients, and in each round a random selection of 20% of the clients participate in the training. For baseline algorithms, we train 500 communication rounds. For FedConcat and FedConcat-ID, we train 480 encoder rounds and 500 classifier rounds to match the communication cost. Since ResNet-50 has huge memory and computation overhead, we set the number of clusters as 2 to constrain memory and computation costs.

New problems arise when training FedConcat with ResNet-50 on CIFAR-100 and Tiny-ImageNet. Firstly, local cluster models tend to overfit since each local cluster witnesses fewer data compared with training on all clients. Secondly, the cluster sizes become quite unbalanced since the label distribution points become more sparse in the high dimensional (100-D or 200-D) space. Some points may be so far from others that they are allocated into a very small cluster. Thirdly, the training process of the final classifier layer is more difficult to converge since there are many more hidden neurons in ResNet-50 than simple CNN.

To address these problems, we increase the weight decay factor to tackle overfitting. Client members of the majority cluster are relocated to force each cluster to be balanced. At the beginning of the post-training stage, the global classifier is initialized with parameters of cluster classifiers to speed up convergence. We discuss these adaptations in detail and conduct ablation studies in Appendix D.8 of our full version (Diao, Li, and He 2023a).

As shown in Table 3, by tackling these issues, our methods achieve higher accuracy than baselines by an average of 8% on CIFAR-100 and 2% on Tiny-ImageNet.

Effect of Clustering

If we concatenate all models from all clients without clustering, when there are a large number of clients, our global model can become very large. Large final model leads to heavy communication and computation costs. Moreover, concatenating all models from all clients can suffer from unstable convergence and low test accuracy, since each client can have limited training samples.

An experiment on CIFAR-10 is illustrated in Figure 3, where we show the test accuracy at each classifier round after training 100 encoder rounds. No clustering means each

Dataset	Partition	FedAvg	FedProx	MOON	FedRS	FedLC	FedConcat	FedConcat-ID
CIFAR-100	$\#C = 2$	8.4%	8.1%	8.0%	7.0%	3.8%	**18.5%**	**16.4%**
	$\#C = 3$	21.6%	18.7%	19.0%	19.2%	12.6%	**34.4%**	**32.9%**
	$p_k \sim Dir(0.1)$	52.4%	51.5%	55.2%	51.8%	50.5%	**61.2%**	**62.1%**
	$p_k \sim Dir(0.5)$	62.0%	61.2%	61.9%	61.9%	61.4%	**66.3%**	**65.6%**
Tiny-ImageNet	$\#C = 2$	3.1%	2.7%	3.0%	3.1%	2.0%	**4.3%**	**4.3%**
	$\#C = 3$	4.9%	5.1%	6.3%	3.3%	1.7%	**11.7%**	**9.6%**
	$p_k \sim Dir(0.1)$	40.8%	40.8%	40.6%	39.7%	39.9%	**43.1%**	**42.6%**
	$p_k \sim Dir(0.5)$	44.0%	44.2%	44.1%	43.6%	43.9%	**44.3%**	43.8%

Table 3: Experimental results on CIFAR-100 and Tiny-ImageNet with ResNet-50. We tune the weight decay among $\{0.00001, 0.001, 0.002, 0.005\}$ for all algorithms. We present the average of the last 10 rounds.

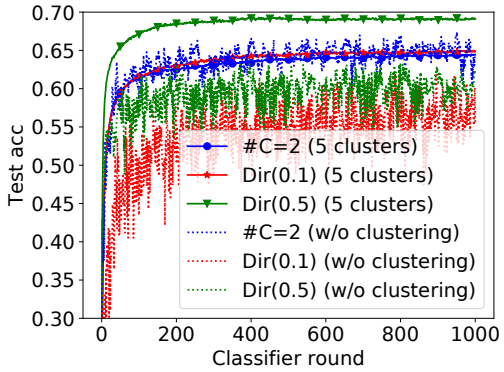

Figure 3: Training curves with clustering versus without clustering on CIFAR-10 (40 clients).

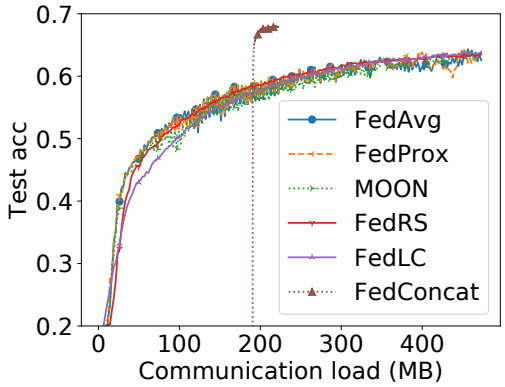

Figure 4: Comparing with baselines on the final global model of FedConcat on CIFAR-10 (40 clients, $\#C = 2$).

client trains a model for server to concatenate, where each client model is only trained with few samples and prone to overfit. Since those models are not well-trained, concatenating their encoders can hardly extract good features. From Figure 3, we can observe that clustering not only reduces communication cost, but also effectively improves model quality with more samples.

FedConcat concatenates small models into a large model. As a baseline, we directly train prior FL algorithms on the model with equivalent size to the concatenated model of FedConcat. The training curves on CIFAR-10 are shown in Figure 4, which illustates that FedConcat still keeps its advantage when compared with the concatenated model.

Comparing with Other Clustered FL

In this section, we employ other clustered FL algorithms during our clustering stage. We conduct experiments with three clustering-based methods including IFCA (Ghosh et al. 2020), recently proposed FedSoft (Ruan and Joe-Wong 2022) and FeSEM (Long et al. 2023). The results for the CIFAR-10 dataset are presented in Table 4. It can be observed that both FedConcat and FedConcat-ID outperform other clustering strategies. FeSEM incorporates an additional proximal loss term during local training, which results in an extra computational burden similar to FedProx. Both IFCA and FedSoft entail multiple times of communication cost as all cluster models are transferred to clients in each round. Thus, the clustering strategies of FedConcat and FedConcat-ID prove to be both effective and efficient.

Partition	Best baseline	FedConcat	FedConcat-ID
$\#C = 2$	54.5%	**56.9%**	**56.5%**
$\#C = 3$	60.4%	**62.0%**	**61.8%**
$p_k \sim Dir(0.1)$	56.1%	**57.7%**	**56.9%**
$p_k \sim Dir(0.5)$	63.3%	**64.2%**	**63.7%**

Table 4: Comparing with other clustering strategies (IFCA, FedSoft and FeSEM) on CIFAR-10.

Conclusion

In this paper, we propose to alleviate the accuracy decay induced by label skews in FL through concatenation. We show that in most cases, our methods can significantly outperform various state-of-the-art FL algorithms with smaller communication costs. FedConcat can alleviate accuracy decay because it divides the hard problem (training a model among all clients under extreme label skews) into various easy problems (training one model within each cluster under alleviated label skews). Then it collects clues of easy problems (i.e., extracted features) to solve the hard original problem. Our approach brings new insights to the FL community to look for other aggregation approaches instead of averaging.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-018). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

References

Bahng, H.; Chun, S.; Yun, S.; Choo, J.; and Oh, S. J. 2020. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, 528–539. PMLR.

Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C. M.; Konečný, J.; Mazzocchi, S.; McMahan, B.; Overveldt, T. V.; Petrou, D.; Ramage, D.; and Roselander, J. 2019. Towards Federated Learning at Scale: System Design. In *SysML*.

Dai, Z.; Low, B. K. H.; and Jaillet, P. 2020. Federated Bayesian optimization via Thompson sampling. *Advances in Neural Information Processing Systems*, 33.

Diao, Y.; Li, Q.; and He, B. 2023a. Exploiting Label Skews in Federated Learning with Model Concatenation. arXiv:2312.06290.

Diao, Y.; Li, Q.; and He, B. 2023b. Towards Addressing Label Skews in One-Shot Federated Learning. In *The Eleventh International Conference on Learning Representations*.

Ghosh, A.; Chung, J.; Yin, D.; and Ramchandran, K. 2020. An Efficient Framework for Clustered Federated Learning. *ArXiv*, abs/2006.04088.

Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; and Ramage, D. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

He, C.; Annavaram, M.; and Avestimehr, S. 2020. Group Knowledge Transfer: Federated Learning of Large CNNs at the Edge. *Advances in Neural Information Processing Systems*, 33.

Hsu, T.-M. H.; Qi, H.; and Brown, M. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.

Johnson, J. M.; and Khoshgoftaar, T. M. 2019. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1): 1–54.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*.

Kaissis, G. A.; Makowski, M. R.; Rückert, D.; and Braren, R. F. 2020. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 1–7.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for on-device federated learning. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2021. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079*.

Li, Q.; He, B.; and Song, D. 2021. Model-Contrastive Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Li, Q.; Wen, Z.; and He, B. 2020. Practical Federated Gradient Boosting Decision Trees. In *AAAI*, 4642–4649.

Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; and He, B. 2019a. A Survey on Federated learning systems: Vision, hype and reality for data privacy and protection. *arXiv preprint arXiv:1907.09693*.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2019b. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020a. Federated optimization in heterogeneous networks. In *MLSys*.

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2020b. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations*.

Li, X.-C.; and Zhan, D.-C. 2021. Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 995–1005.

Liu, Y.; Kang, Y.; Xing, C.; Chen, T.; and Yang, Q. 2020. A Secure Federated Transfer Learning Framework. *IEEE Intelligent Systems*.

Lloyd, S. P. 1982. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28: 129–136.

Long, G.; Xie, M.; Shen, T.; Zhou, T.; Wang, X.; and Jiang, J. 2023. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1): 481–500.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; et al. 2016. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Ruan, Y.; and Joe-Wong, C. 2022. Fedsoft: Soft clustered federated learning with proximal local updating. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8124–8131.

Shwartz-Ziv, R.; and Tishby, N. 2017. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.

Voigt, P.; and Von dem Bussche, A. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*.

Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; and Khazaeni, Y. 2020a. Federated Learning with Matched Averaging. In *International Conference on Learning Representations*.

Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; and Poor, H. V. 2020b. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems*, 33.

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

Wu, J.; Zhang, Q.; and Xu, G. 2017. Tiny imagenet challenge. *Technical report*.

Wu, Y.; Cai, S.; Xiao, X.; Chen, G.; and Ooi, B. C. 2020. Privacy preserving vertical federated learning for tree-based models. *Proceedings of the VLDB Endowment*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2): 1–19.

Zhang, J.; Li, Z.; Li, B.; Xu, J.; Wu, S.; Ding, S.; and Wu, C. 2022. Federated Learning with Label Distribution Skew via Logits Calibration. In *International Conference on Machine Learning*, 26311–26329. PMLR.