Behavior Mimics Distribution: Combining Individual and Group Behaviors for Federated Learning

Hua Huang[1] , **Fanhua Shang**[1,2*] , **Yuanyuan Liu**[1] and **Hongying Liu**[1*]
[1] Key Lab of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, China
[2] Peng Cheng Lab, Shenzhen, China
{hhuang, fhshang, yyliu, hyliu}@xidian.edu.cn

Abstract

Federated Learning (FL) has become an active and promising distributed machine learning paradigm. As a result of statistical heterogeneity, recent studies clearly show that the performance of popular FL methods (e.g., FedAvg) deteriorates dramatically due to the client drift caused by local updates. This paper proposes a novel Federated Learning algorithm (called **IGFL**), which leverages both ***I**ndividual* and ***G**roup* behaviors to mimic distribution, thereby improving the ability to deal with heterogeneity. Unlike existing FL methods, our IGFL can be applied to both client and server optimization. As a by-product, we propose a new *attention-based federated learning* in the server optimization of IGFL. To the best of our knowledge, this is the first time to incorporate attention mechanisms into federated optimization. We conduct extensive experiments and show that IGFL can significantly improve the performance of existing federated learning methods. Especially when the distributions of data among individuals are diverse, IGFL can improve the classification accuracy by about 13% compared with prior baselines.

1 Introduction

Federated learning is originally designed to allow massive mobile devices to collaboratively train a global statistical model. In recent years, it has emerged as a new paradigm for large-scale distributed machine learning without exchanging privacy data. The traditional distributed methods require that individuals upload their local data to a cloud parameter server (PS), and the server updates the model, where individuals can be remote devices or siloed data centers, such as banks and hospitals. An urgent problem is that most individuals are not willing to communicate their personal data, such as private photos or personal financial information, to an untrusted third party [McMahan *et al.*, 2017].

General federated learning first distributes the aggregation model to each client by the parameter server. The clients update the model multiple times locally, and then communicate the model updates to the server. During the whole process, FL requires data not to be shared, which ensures the privacy of the clients. Statistical heterogeneity is a key and tough challenge for FL, which is usually not considered in distributed machine learning. Individuals with unique attributes make the collected data examples have different statistical distributions. Unfortunately, many recently proposed distributed machine learning methods are based on the independent and identically distributed (I.I.D.) assumption. This makes most of them directly applied to FL often ineffective [Li *et al.*, 2020a]. In addition, similar to distributed machine learning, federated learning also suffers from expensive communication [Konecny *et al.*, 2016]. With the development of modern mobile devices, the computational cost of the devices are usually far less than the communication cost [Li *et al.*, 2020a]. A common communication-efficient solution is to use Local SGD [Stich, 2019; Lin *et al.*, 2020]. This method allows clients to perform multiple local updates instead of once, and then communicate the results to the central server.

The FL algorithm based on Local SGD is the widely-used FedAvg [McMahan *et al.*, 2017]. However, the multiple local updates keep the client away from the global optimum. In extreme cases, each client reaches the local optimum, and then the server aggregates them. This is equivalent to ont-shot averaging, which does not work in non-IID setting [Reddi *et al.*, 2021]. Subsequently, researchers proposed a variety of new and improved algorithms. These methods can be roughly divided into two categories: i) *client optimization*. The typical algorithm is SCAFFOLD [Karimireddy *et al.*, 2020], which uses control variates at the stage of local updates, to alleviate the client drift. Inspired by variance reduction [Johnson and Zhang, 2013; Shang *et al.*, 2021], [Liang *et al.*, 2019] proposed a similar method. ii) *server optimization*. These methods involve a server aggregation stage with momentum acceleration on the server as in FedAvgM [Hsu *et al.*, 2019] or an adaptive optimizer [Kingma and Ba, 2015] on the server as in FedAdam [Reddi *et al.*, 2021]. This category of algorithms regard the aggregation of the information from various individuals as a pseudo-gradient-based optimization process, which is called server optimization. In FedAvg, its server learning rate is always 1, which makes it lack of flexibility in server optimization.

Since the data is not shared, only the distributions of individuals which are different from the distribution of the whole

*Corresponding authors.

group, are used for local updates. Server aggregation is actually a synchronization function that can eliminate the impact of this difference to a certain extent, depending on the frequency of communication. Therefore, this raises a question: *Is there a better way to integrate individual distribution into group distribution instead of just relying on synchronization?* We answer this question in the **affirmative** and propose a new federated learning algorithm, called ***I**ndividual and **G**roup **F**ederated **L**earning* (**IGFL**). IGFL leverages the behaviors (e.g., updates) of individuals and groups to mimic the corresponding distributions, and digests them in client optimization and server optimization. In particular, inspired by the success of attention mechanisms [Bahdanau *et al.*, 2015], we investigate the method of incorporating attention into federal learning and applied it to the server optimization of IGFL.

1.1 Our Contributions

We summarize our main contributions as follows.

• In order to reduce the adverse impacts of client drift caused by local updates, we propose a novel and unified federated learning algorithm (called **IGFL**). IGFL has an insight into the interplay between the behaviors and distributions of individuals and groups, that help to improve both client optimization and server optimization. IGFL with only our client or server optimization is called IGFL-C or IGFL-S.

• Moreover, taking the behaviors of individuals and groups into account for server optimization, and inspired by the success of attention mechanisms, we also propose a new ***Attention-based Federated Learning*** (AFL) scheme. To the best of our knowledge, this is the first time to incorporate attention into federated optimization.

• Finally, We conduct extensive experiments on the CIFAR10 and EMNIST data sets, and all the results show that the proposed algorithms exhibit superior performance to other algorithms in various situations. Especially, our algorithm can improve the classification accuracy by about 13% compared with prior baselines (e.g., FedAvg) in highly heterogeneous (non-IID) settings.

2 Related Work

FedAvg [McMahan *et al.*, 2017] is a standard federated optimization algorithm, but it is tricky to deal with heterogeneous data. Recently, in order to better deal with non-IID data, researchers have proposed many improved methods. FedProx proposed in [Li *et al.*, 2020b] add a proximal term to the objective for stable updates. [Zhao *et al.*, 2018] presented a sharing strategy to improve performance, but it violated the basic requirement of federated learning. FedAvgM is a method of adding momentum on the server, proposed by [Hsu *et al.*, 2019], and the authors also proposed a new way, which rely on Dirichlet distribution, to generate measurable federated data sets. SCAFFOLD [Karimireddy *et al.*, 2020] uses control variables to alleviate client drift, and it can be viewed as employing the idea of variance reduction on the client. Similar to SCAFFOLD, VRL-SGD [Liang *et al.*, 2019] also utilizes the method of variance reduction, but does not support client sampling. FedAdam [Reddi *et al.*, 2021] introduces an adaptive optimization method on the server side. Unlike the synchronous method, [Chen *et al.*, 2020] proposed an computationally efficient and asynchronous online method.

Algorithm 1 IGFL

Input: P, R.
Initialization:w_{ps}^0, $\Delta w_k^0 = 0, k \in [P]$.
Output: w_{ps}^R.

1: **for** $r = 0, ..., R-1$ **do**
2: Sample subset $\mathcal{S}$ from clients;
3: **for** each client $i \in \mathcal{S}$ **in parallel do**
4: Δw_i^{r+1} =**IGFL-client**$(w_{ps}^r,\ i,\ \Delta w_i^r)$;
5: **end for**
6: w_{ps}^{r+1} =**IGFL-server**$(\Delta w_i^{r+1}, \Delta w_i^r | i \in \mathcal{S}\})$;
7: **end for**
8: **return** w_{ps}^R.

3 Individual and Group Federated Learning

For general federated learning problems, they can be expressed as the following problem:

$$\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{P} \sum_{i=1}^{P} f_i(w), \tag{1}$$

where $f_i(w) := \mathbb{E}_{x \sim \mathcal{P}_i}[f_i(w, x)]$ is the local loss function of the i-th client, P is the total number of clients, and $\mathcal{P}_i$ denotes the distribution for the i-th local data. We assume f_i is smooth. In this paper, we focus on the non-IID setting, where $\mathcal{P}_i$ may be very different (i.e., $\mathbb{E}_{\mathcal{P}_i}[f_i(w)] \neq f(w)$).

Our proposed **IGFL** algorithm is described in Algorithm 1. It includes two important schemes, *IGFL-client* (Line 4) and *IGFL-server* (Line 6), which are shown in Algorithms 2 and 3, respectively. If we formulate the main update rules in Lines 4 and 6 into Eqs. (4) and (5), IGFL degenerates into FedAvg. We refer to the algorithms that only apply the client or server optimization scheme by IGFL-C (see Section 3.2 for details) and IGFL-S (see Section 4 for details), respectively. The idea behind IGFL is to make full use of both individual and group behaviors to approximate distribution $\mathcal{P}$ in client optimization and server optimization.

3.1 Review SGD and Local SGD

To motivate the algorithms in this paper, we revisit the difference between SGD and Local SGD to illustrate the shortcomings of Local SGD. We assume that all clients are activated and the results obtained are easily extended to partly activation. For the i-th client in the r-th round, after pulling the global model (i.e., $w_i^r \leftarrow w_{ps}^r$), mini-batch SGD [Shang *et al.*, 2020] can be written in the following form:

$$w_i^{r+1} = w_i^r - \eta_l g_i(w_i^r), \tag{2}$$

where η_l is the client learning rate, w_{ps}^r and w_i^r are the parameters of the server and the i-th client in the r-th round, respectively. By aggregating w_i^{r+1} on the server, we have

$$w_{ps}^{r+1} = \frac{1}{P} \sum_{i=1}^{P} w_i^{r+1} = w_{ps}^r - \frac{\eta_l}{P} \sum_{i=1}^{P} g_i(w_{ps}^r), \tag{3}$$

Algorithm 2 IGFL-client

Input: w_{ps}^r, i, Δw_i^r.
Parameters: T, η_l.
Output: Δw_i^{r+1}.

$w_i^{r,0} = w_{ps}^r$; // pull global parameters
$\Delta w_{ps} = w_{ps}^r - w_{ps}^{r-1}$;
for $t = 0, ..., T-1$ **do**
  Compute gradient $g_i(w_i^{r,t})$;
  $\Delta_I = -\eta_l g_i(w_i^{r,t})$, $\Delta_G = \frac{1}{|\mathcal{S}|}(\Delta_I - \frac{1}{T}\Delta w_i^r) + \frac{1}{T}\Delta w_{ps}$;
  $w_i^{r,t+1} = w_i^{r,t} + \Delta_I + \Delta_G$;
end for
$\Delta w_i^{r+1} = w_i^{r,T} - w_{ps}^r$;
return Δw_i^{r+1}.

where P is the number of client. While Local SGD has the following form in the i-th client:

$$w_i^{r+1} = w_{ps}^r - \eta_l \sum_{t=0}^{T-1} g_i(w_i^{r,t}), \tag{4}$$

where T is the number of local steps. After aggregation,

$$w_{ps}^{r+1} = w_{ps}^r - \frac{\eta_l}{P} \sum_{i=1}^{P} \sum_{t=0}^{T-1} g_i(w_i^{r,t}), \tag{5}$$

where $w_i^{r,t}$ denotes the parameters of i-th client in the r-th round at t-th local step. Thanks to aggregation without delay, each update of SGD is based on the entire example set $x_{\sim\mathcal{P}}$. Unfortunately, the local updates of Local SGD depend only on the local example set $x_{\sim\mathcal{P}_i}$, resulting in the lack of interacting with other individuals in time. It tends to have a large deviation due to walking alone for a long time, especially as the number of local updates increases.

3.2 Use of Individual and Group Information

In order to make corrections, we want it to make full use of group information at every step forward, not just its own individual information. Thus, we have the following rules:

$$w_i^{r+1} = w_{ps}^r - \eta_l \sum_{t=0}^{T-1} g_i^{t*}, g_i^{t*} = \frac{1}{P}[g_i(w_i^{r,t}) + \sum_{k\neq i}^{P} g_k(w_k^{r,t})]. \tag{6}$$

However, as a result of the basic demands for privacy in FL, the individuals cannot get $\sum_{k\neq i}^{P} g_k(w_k^{r,t})$. We define the approximation of individual and group behaviors as follows:

$$\Delta w_{ps}^r := w_{ps}^r - w_{ps}^{r-1}, \quad \Delta w_i^r := w_i^r - w_i^{r-1}, \tag{7}$$

which are used to simulate the responses of individual distribution $\mathcal{P}_i$ and group distribution $\mathcal{P}$, respectively, and $\Delta w_{ps}^r = \frac{1}{P}\sum_{i=1}^{P} \Delta w_i^r$. We can assume that the gradient $g_i(w_i^{r,t})$ does not change quickly because $f_i(w)$ is smooth, and $g_k(w_i^{r,t}) \approx g_k(w_i^{r-1,t})$, similar to [Karimireddy *et al.*, 2020]. For the update rules (4) and (5), we have

$$\sum_{t=0}^{T-1} \sum_{k\neq i}^{P} g_k(w_k^{r,t}) \approx \frac{P}{-\eta_l}\Delta w_{ps}^r - \frac{1}{-\eta_l}\Delta w_i^r. \tag{8}$$

Now, plugging Eq. (8) back to Eq. (6), we get the gradient with respect to the approximate distribution of the group:

$$\begin{aligned}\hat{g}_i^t &= \frac{1}{P}[g_i(w_i^{r,t}) + \frac{P}{-\eta_l T}\Delta w_{ps}^r - \frac{1}{-\eta_l T}\Delta w_i^r] \\ &= \frac{1}{P}(g_i(w_i^{r,t}) - \frac{1}{-\eta_l T}\Delta w_i^r) + \frac{1}{-\eta_l T}\Delta w_{ps}^r. \end{aligned} \tag{9}$$

Note that $\hat{g}_i^t$ represents the gradient that only depends on group behavior when updating locally. Furthermore, we hope that our algorithm not only depends on group behavior, but also ensures that individual specificity is not diluted. Therefore, in each step of the local update, clients also examine themselves sufficiently. Specifically, $g_i^t = \hat{g}_i^t + g_i(w_i^{r,t})$. The detailed algorithm is given in Algorithm 2.

Compared with FedAvgM. If $\sum_{t=0}^{T-1}\sum_{k\neq i}^{P} g_k(w_k^{r,t})$ is added directly after T local updates as additional distribution information, instead of spreading them to each local update, the update rule becomes:

$$w_i^{r+1} \leftarrow w_i^{r+1} + \Delta w_{ps}^r - \frac{1}{P}\Delta w_i^r. \tag{10}$$

Obviously, it is equivalent to injecting information directly on the server: $w_{ps}^{r+1} = \frac{1}{P}\sum_{i=1}^{P} w_i^{r+1} + \frac{P-1}{P}\Delta w_{ps}^r$. In terms of the rule, this is similar to FedAvgM proposed in [Hsu *et al.*, 2019], which has the following form: $v \leftarrow \beta v + \Delta w_{ps}^{r+1}$, $w_{ps}^{r+1} = w_{ps}^r + v$. Note that the additional information injected by FedAvgM is not limited to the group behaviors of the previous round, but also those in the past. Besides, it will suffer from some major drawbacks if the information is injected at once: i) *Volume*, $||\Delta w_{ps}^r||$ is much larger than the ordinary local update, which increases the uncertainty. ii) *Timeliness*, more importantly, the internal gradient calculation still only contains individual behaviors. The timeliness of approximate group behaviors is further weakened. iii) *Specificity*, the individual behaviors (i.e., Δw_j^r) are not utilized in this method.

Compared with SCAFFOLD. Amazing and marvellous, $\hat{g}_i^t$ has a similar form as the update of SCAFFOLD [Karimireddy *et al.*, 2020]. Compared with SCAFFOLD, there is a subtle difference that IGFL-C changes the coefficient of $(g_i(w_i^{r,t}) - \frac{1}{-\eta_l T}\Delta w_i^r)$ from 1 to $1/P$. Formally, $1/P$ comes from Eqs. (8) and (6). As described in [Defazio *et al.*, 2014], a variance reduction method is to use the estimator $\tilde{\nabla}_\xi := \xi(X-Y)+\mathbb{E}Y$ to approximate $\mathbb{E}X$, where $\xi \in [0,1]$ and Y is highly correlated with X. We have $\mathbb{E}\tilde{\nabla}_\xi = \xi\mathbb{E}X + (1-\xi)\mathbb{E}Y$, $\text{Var}(\tilde{\nabla}_\xi) = \xi^2[\text{Var}(X) + \text{Var}(Y) - 2\,\text{Cov}(X,Y)]$. Here, we set $g_i(w) = X$ and $\Delta w_i = Y$. SCAFFOLD uses $\xi = 1$ to obtain an unbiased estimator. IGFL-C uses $\xi = 1/P$ and has a non-zero bias. But the variance of IGFL-C is $1/P^2$ times the one of SCAFFOLD. In fact, due to the statistical heterogeneity, $\mathbb{E}_{\mathcal{P}_i}[g_i(w)] \neq \mathbb{E}_{\mathcal{P}}[g(w)]$, both of SCAFFOLD and IGFL-C are biased. Moreover, IGFL-C uses behaviors to simulate distributions, which can correct certain biases.

4 Attention-based Federated Learning

The innovation of IGFL-client is mainly to apply the idea of simulating the responses of distributions by combining individual and group behaviors to the client optimization. We

answer the following question in the **affirmative**: *Whether this idea can also be applied to the server optimization to improve the ability of tackling statistical heterogeneity?* In this section, inspired by the attention mechanism, we introduce a novel ***Attention-based Federated Learning*** (AFL) scheme as a tool for server optimization of IGFL. In essence, it measures the similarity between individual and group behaviors, or the similarity between two rounds of behaviors, and gives each client a new weight for summing them together. The attention mechanism has been widely used in many fields such as natural language processing [Bahdanau *et al.*, 2015] and computer vision [Mnih *et al.*, 2014], and it is playing an increasingly important role. Next we will introduce how to use the attention mechanism innovatively to improve the capacity of federated learning to accommodate heterogeneity. To the best of our knowledge, this is the first time to apply attention mechanism to federated optimization.

As described in [Vaswani *et al.*, 2017], the attention mechanism can be regarded as a mapping function about a set of key-value pairs and a query Q. Specifically, the similarity between Q and each key is calculated to obtain a set of scores, which are used as weights. The output is a weighted sum of $values$. At the beginning of the r-th round of server optimization, the central server obtains the individual behaviors of selected clients $\{\Delta w_i^{r+1}|i \in \mathcal{S}\}$. In a general FL algorithm, the update rule of this round is the average of these behaviors: $w_{ps}^{r+1} = w_{ps}^r + \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \Delta w_i^{r+1}$. In the proposed AFL, we treat these behaviors as a set of values V. By defining specific keys K and queries Q, we can get the attention output $\Delta\hat{w}_i^{r+1}$. After that, the new parameters w_{ps}^{r+1} are obtained by a mapping $\mathcal{M}(\{\Delta\hat{w}_i^{r+1}|i \in \mathcal{S}\}, w_{ps}^r)$, such as $w_{ps}^{r+1} = w_{ps}^r + \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \Delta\hat{w}_i^{r+1}$. We compute $\Delta\hat{w}_i^{r+1}$ according to the following update rule:

$$\Delta\hat{w}_i^{r+1} = \sum_{j\in\mathcal{S}} \alpha_{ij}^r \Delta w_j^{r+1},\ \alpha_{ij}^r = \frac{e^{\psi(Q_i^r, \Delta w_j^{r+1})}}{\sum_\tau e^{\psi(Q_i^r, \Delta w_\tau^{r+1})}}, \quad (11)$$

where $\psi(\cdot)$ is a similarity measurement function (e.g., we use dot product in this paper). Note that we set $K := V = \{\Delta w_i^{r+1}|i \in \mathcal{S}\}$, which is consistent with the attention mechanism in other areas such as [Bahdanau *et al.*, 2015]. Regarding the selection of Q, we provide three different strategies, which are discussed in detail below.

4.1 Self-Attention Federated Learning

In this subsection, we introduce a new self-attention federated learning scheme, where $Q := K = V = \{\Delta w_i^{r+1}|i \in \mathcal{S}\}$, similar to [Vaswani *et al.*, 2017]. Hence, the scores α_{ij}^r capture the similarity between individual i and individual j by measuring the similarity of their behaviors in this round. This means that the individual that is more similar to individual i should have more greater weight when representing individual i. The update rules can be formulated as follows:

$$\Delta\hat{w}_i^{r+1} = \sum_{j\in\mathcal{S}} \alpha_{ij}^r \Delta w_j^{r+1},\ \alpha_{ij}^r = \frac{e^{\psi(\Delta w_i^{r+1}, \Delta w_j^{r+1})}}{\sum_\tau e^{\psi(\Delta w_i^{r+1}, \Delta w_\tau^{r+1})}}.$$

Algorithm 3 IGFL-server

Input: $\{\Delta w_i^{r+1}, \Delta w_i^r | i \in \mathcal{S}\}$.
Output: w_{ps}^{r+1}.

$\Delta w_{ps}^{r+1} = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \Delta w_i^{r+1}$;
for each $i \in \mathcal{S}$ **do**
$q_i \leftarrow \begin{cases} \Delta w_i^{r+1}, & \text{Option I for self-attention} \\ \Delta w_{ps}^{r+1}, & \text{Option II for global-attention} \\ \Delta w_j^r, & \text{Option III for time-attention} \end{cases}$
$\alpha_{ij}^r = \frac{e^{\psi(q_i, \Delta w_j^{r+1})}}{\sum_\tau e^{\psi(q_i, \Delta w_\tau^{r+1})}}$;
$\Delta\hat{w}_i^{r+1} = \sum_{j\in\mathcal{S}} \alpha_{ij}^r \Delta w_j^{r+1}$;
end for
$w_{ps}^{r+1} = w_{ps}^r + \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} \Delta\hat{w}_i^{r+1}$;
return w_{ps}^{r+1}.

4.2 Global-Attention Federated Learning

The attention score can be calculated not only from the similarity between individuals, but also from the similarity between individuals and groups. We propose a new global-attention federated learning scheme, i.e., $Q = \Delta w_{ps}^{r+1}$. Note that for any $i \in \mathcal{S}$, Q_i does not change, i.e., there is only one query. As a result, for any $i \in \mathcal{S}$, $\Delta\hat{w}_i^{r+1}$ is equal. The update rules for server optimization become

$$w_{ps}^{r+1} = w_{ps}^r + \sum_{i\in\mathcal{S}} \alpha_i^r \Delta w_i^{r+1},\ \alpha_i^r = \frac{e^{\psi(\Delta w_{ps}^{r+1}, \Delta w_i^{r+1})}}{\sum_\tau e^{\psi(\Delta w_{ps}^{r+1}, \Delta w_\tau^{r+1})}}.$$

The scores measure the similarity between individual and group behaviors. We should focus on an individual if its behavior is similar to the group behavior.

4.3 Time-Attention Federated Learning

The above two methods utilize the similarity between the individuals, or the individuals and the group in the current round. Time-attention FL sets $Q_i = \Delta w_j^{r-1}$ and provides a way to distribute attention by capturing the similarity between the behaviors of the current round and the previous round. One can see that the score α_{ij}^r is decoupled from i and simplified to α_j^r, similar to global-attention FL. Time-attention FL can be formulated as follows:

$$w_{ps}^{r+1} = w_{ps}^r + \sum_{j\in\mathcal{S}} \alpha_j^r \Delta w_j^{r+1},\ \alpha_j^r = \frac{e^{\psi(\Delta w_j^r, \Delta w_j^{r+1})}}{\sum_\tau e^{\psi(\Delta w_j^r, \Delta w_\tau^{r+1})}}.$$

It can be interpreted that if the behaviors of an individual in two rounds are very similar, this means that the group behavior has little impact on individual correction, and this also implies that the individual is similar to the group.

5 Experiments

In this section, we demonstrate empirical evaluation of the proposed algorithms. We first verify the effectiveness of our algorithm by training the model using convolutional networks on the CIFAR10 [Krizhevsky and Hinton, 2009] and EMNIST [Cohen *et al.*, 2017] data sets. Here we use the same

Cross-silo	$E=1$	$E=5$
FedAvg	75.89	72.16
FedAvgM	76.90	72.30
SCAFFOLD	76.49	73.78
FedAdam	76.29	71.22
IGFL-C (ours)	77.25	74.69
IGFL-S (ours)	76.17	73.61
IGFL (ours)	**78.08**	**75.56**

Table 1: Comparison of the average testing accuracies (%) over the last 10% rounds of each algorithm on CIFAR10 in the cross-silo setting after 5,000 or 3,000 communication rounds, which corresponds to $E=1$ and $E=5$, respectively. For IGFL-S and IGFL, we use the time-attention scheme to achieve the best performance.

network structure as in [McMahan *et al.*, 2017] and [Reddi *et al.*, 2021], respectively. We also conduct extensive experiments to further discuss IGFL. We implement all experiments in Ray [Moritz *et al.*, 2018] based Python, which is a flexible, high-performance distributed execution framework.

5.1 Setup

Data partitions. The non-IID populations are generated in two schemes: i) Sort-and-partition [McMahan *et al.*, 2017]. Each client has two shares with different labels, and each share is randomly selected from data partitions sorted by labels. ii) Dirichlet distribution [Hsu *et al.*, 2019]. The training examples in each client are extracted by class following a categorical distribution generated by a Dirichlet distribution, $\boldsymbol{c} \sim Dir(\rho \boldsymbol{q})$, where $\rho > 0$ is a concentration parameter adjusting the heterogeneity among clients and $\boldsymbol{q}$ is a prior probability, assuming a uniform distribution.

Hyperparameter tuning and methods. For both CIFAR10 and EMNIST, we used the following parameters: batch size $B = 100$, local epoch $E = \{1, 5\}$, and client selection rate $C = \{0.1, 1\}$, without step decay. We set the grid search range of client learning rate by $\eta_l \in \{10^{-3}, 3 \times 10^{-3}, ..., 10^{-1}, 3 \times 10^{-1}\}$. We fixed the server learning rate to 1, except for FedAadam. We use the following methods as compared algorithms: FedAvg [McMahan *et al.*, 2017], FedAvgM [Hsu *et al.*, 2019], SCAFFOLD [Karimireddy *et al.*, 2020], and FedAdam [Reddi *et al.*, 2021]. Specifically, we set $\beta = 0.9$ for FedAvgM, and adjust $\tau \in \{0.1, 0.01\}$ to achieve the best performance for FedAdam.

5.2 Main Results

We conduct extensive experiments to evaluate the performance of the proposed algorithms in the following two settings: i) *Cross-silo setting*, in which we set $P=10$, the selection rate $C = 1$, and adopt the sort-and-partition scheme. ii) *Cross-device*, in which we set $P=100$, $C=0.1$, and use the Dirichlet distribution scheme.

For the cross-silo setting, Table 1 shows the performance comparison of all the algorithms on CIFAR10 for visual classification tasks, when $E = 1$ or $E = 5$. Meanwhile, we give the test accuracy curve along with the number of rounds, as shown in Figure 1. Experimental results show that the proposed algorithms exhibit superior performance to other algorithms. Note that when encountering a larger number of local epochs, (i.e., $E = 5$), the performance of FedAvgM degenerates and is only slightly better than that of FedAvg, while IGFL still has a significantly improvement over FedAvg. We will discuss this phenomenon in detail in Section 5.3.

	Cross-device	$\rho=1000$	$\rho=1$	$\rho=0.1$
$E=1$	FedAvg	82.03	75.22	68.62
	FedAvgM	82.21	77.34	70.12
	SCAFFOLD	80.99	75.97	–
	FedAdam	82.55	76.21	72.21
	IGFL-C (ours)	83.56	79.51	78.71
	IGFL-S (ours)	83.17	79.14	73.09
	IGFL (ours)	**84.01**	**81.33**	**79.45**
$E=5$	FedAvg	82.21	73.29	63.67
	FedAvgM	82.75	73.62	67.10
	SCAFFOLD	81.83	71.84	–
	FedAdam	82.82	75.50	67.07
	IGFL-C (ours)	83.10	78.50	72.31
	IGFL-S (ours)	82.51	73.63	66.79
	IGFL (ours)	**84.13**	**78.63**	**76.38**

Table 2: Comparison of the average testing accuracy (%) over the last 10% rounds of each algorithm on CIFAR10 in the cross-device setting after 10,000 or 4,000 communication rounds, which corresponds to $E = 1$ and $E = 5$, respectively. For IGFL-S and IGFL, we use the global-attention scheme.

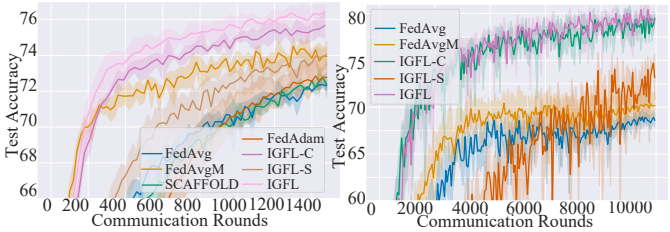

Figure 1: Test accuracies vs. the number of rounds of all the algorithms on CIFAR10 in the cross-silo (left) and cross-device (right) settings (here $E=1$ and $\rho=0.1$, best viewed in color).

In the cross-device setting, Tables 2 and 3 show the corresponding experimental results on CIFAR10 and EMNIST, respectively. The proposed methods achieve impressive improvements in terms of accuracy over other methods. In particular, it is exciting to see our algorithm improves accuracy by nearly **13%** over the baseline method, (i.e., FedAvg), on CIFAR10 for the highly skewed non-IID case (e.g., $\rho = 0.1$), while the improvement of other federated optimization algorithms (e.g., FedAvgM, FedAdam) is only about **4%**.

5.3 Effectiveness of Amortization Against FedAvgM

We design and run experiments to discuss the relationship between IGFL-C and FedAvgM to illustrate that IGFL-C can significantly reduce client drift by spreading the approximate group distribution information to each local update, rather than at the end of each round. We change T, which depends on B and E. Specifically, we set $B = \{20, 100\}$, $E = \{1, 5\}$. The experimental results are shown in Table 4. It can be found

Cross-device	$\rho = 1$	$\rho = 0.1$
FedAvg	77.72	70.83
FedAvgM	78.07	75.37
SCAFFOLD	78.08	68.20
FedAdam	77.90	71.17
IGFL-C (ours)	78.96	72.61
IGFL-S (ours)	78.88	74.63
IGFL (ours)	**81.17**	**77.27**

Table 3: Comparison of the average testing accuracies (%) over the last 10% rounds of each algorithm on EMNIST in the cross-device setting after 1,500 communication rounds. For IGFL-S and IGFL, we use the self-attention scheme.

T (B, E)	FedAvgM	IGFL-C	δ
50 ($B=100$, $E=1$)	76.90	77.25	0.35
250 ($B=20$, $E=1$)	71.83	73.35	1.52
250 ($B=100$, $E=5$)	72.30	74.69	2.39
1250 ($B=20$, $E=5$)	68.54	71.54	3.00

Table 4: Comparison of the testing accuracies (%) of FedAvgM and IGFL-C as varying T in the cross-silo setting.

that as the number of local updates increases, the performance difference between them becomes more and more obvious, which shows the advantages of the amortization strategy used in IGFL-C when encountering a large local epoch number.

5.4 Comparison of Attention-based Methods

We first compare the performance of the three proposed attention schemes, as shown in Table 5. The experimental results suggest that specific settings may require specific attention to achieve the best performance. For example, in the case of no client sampling, time-attention performs very well, because under the sampling conditions, the previous behaviors saved by different clients correspond to different rounds. Global-attention has excellent performance on CIFAR10 (10 classes), but not well on EMNIST (62 classes), which implies that its performance may be affected by the number of classes. More experiments are needed for conclusions.

Moreover, the visualization of how the behaviors mimic the distributions is given in Figure 2. We set two specific populations for CIFAR10: i) random population by using the standard sort-and-partition, and ii) paired population, in which every two clients have the same label distribution. We generate 50 different random populations for experiments, and calculate the matching rate between client pairs with the same label and darker coordinate pairs on the heat map, which is up to 96%. Experimental results show that self-attention can indeed capture the similarity of distribution between clients, by measuring the similarity of individual behaviors.

6 Conclusions

In this paper, we proposed a novel federated learning algorithm (called IGFL) with both a new client optimizer and a new server optimizer to alleviate the statistical heterogeneity for federated learning. It leverages individual behavior as an estimate of the data distribution of individual, and complements

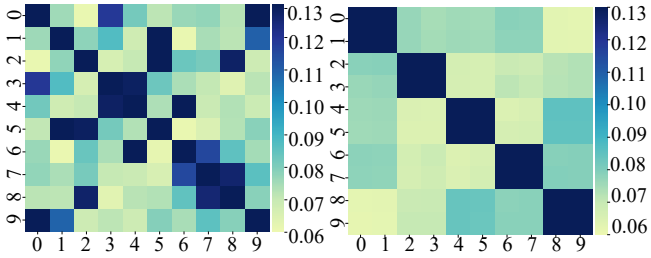

Figure 2: Visualization of how self-attention works, where the attention scores are the average of all (1,000 rounds) scores. The populations corresponding to random example (left) and paired example (right) are {(0,9), (1,5), (2,8), (3,0), (4,3), (5,2), (6,4), (7,6), (8,7), (9,1)} and {(0,1), (0,1),(2,3), (2,3),..., (8,9), (8,9)}, respectively, where (i, j) denotes the client assigned the i-th and j-th labels.

Setting	E	ρ	GA	SA	TA
CIFAR10 silo	$E=1$	–	77.48	76.46	**78.08**
	$E=5$	–	74.50	73.91	**75.56**
CIFAR10 device	$E=1$	1000	84.01	83.30	**84.17**
		1	**81.33**	80.07	79.95
		0.1	**79.45**	79.03	76.83
	$E=5$	1000	**84.13**	83.16	80.06
		1	**78.63**	78.56	77.94
		0.1	**76.38**	75.92	74.89
EMNIST device	$E=1$	1	78.51	**81.17**	78.73
		0.1	70.45	**77.27**	72.57

Table 5: Testing accuracies (%) of our three attention schemes in different settings. Here, GA, SA and TA denote the global-attention, self-attention, and time-attention schemes, respectively.

the distribution of groups that are invisible but crucial during local updates, thus reducing the impact of client drift on FL algorithms. Different from other solutions, the technique that behavior mimic distribution can be used in both client optimization and server optimization. As a by-product, we also presented two federated learning algorithms with the proposed client optimizer or server optimizer, which are called IGFL-C and IGFL-S, respectively. Moreover, for our server optimizer, we proposed a federated learning optimization scheme (called AFL) based on the proposed attention mechanisms. The principle behind AFL is that attention captures the similarity of individual distributions. We performed extensive experiments to demonstrate that the proposed algorithms have significant improvements over other recently proposed algorithms, especially in the highly skewed non-IID case.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61876220, 61876221, 61976164, 61836009 and U1701267), the Program for Cheung Kong Scholars and Innovative Research Team in University (No. IRT_15R53), the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048), and the National Science Basic Research Plan in Shaanxi Province of China (No. 2020JM-194).

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

[Chen *et al.*, 2020] Yujing Chen, Yue Ning, Martin Slawski, and Huzefa Rangwala. Asynchronous online federated learning for edge devices with non-iid data. In *IEEE International Conference on Big Data*, pages 15–24, 2020.

[Cohen *et al.*, 2017] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, 2017.

[Defazio *et al.*, 2014] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, volume 27, pages 1646–1654, 2014.

[Hsu *et al.*, 2019] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[Johnson and Zhang, 2013] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, volume 26, pages 315–323, 2013.

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

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[Konecny *et al.*, 2016] Jakub Konecny, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[Li *et al.*, 2020a] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

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

[Liang *et al.*, 2019] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local SGD with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

[Lin *et al.*, 2020] Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local SGD. In *International Conference on Learning Representations*, 2020.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[Mnih *et al.*, 2014] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, volume 27, pages 2204–2212, 2014.

[Moritz *et al.*, 2018] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging AI applications. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 561–577, 2018.

[Reddi *et al.*, 2021] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konecny, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

[Shang *et al.*, 2020] Fanhua Shang, Kaiwen Zhou, Hongying Liu, James Cheng, Ivor Tsang, Lijun Zhang, Dacheng Tao, and Licheng Jiao. VR-SGD: A simple stochastic variance reduction method for machine learning. *IEEE Transactions on Knowledge and Data Engineering*, 32(1):188–202, 2020.

[Shang *et al.*, 2021] Fanhua Shang, Hua Huang, Jun Fan, Hongying Liu, Yuanyuan Liu, and Jianhui Liu. Asynchronous parallel, sparse approximated svrg for high-dimensional machine learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[Stich, 2019] Sebastian U Stich. Local SGD converges fast and communicates little. In *Proceedings of International Conference on Learning Representations*, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[Zhao *et al.*, 2018] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.