Federated Learning for Non-IID Data via Unified Feature Learning and Optimization Objective Alignment

Lin Zhang[1,2] Yong Luo[3] Yan Bai[1,2] Bo Du[3] Ling-Yu Duan[1,2*]
[1] Institute of Digital Media (IDM), Peking University, Beijing, China
[2] Peng Cheng Laboratory, Shenzhen, China
[3] Institute of Artificial Intelligence, School of Computer Science, Wuhan University, Wuhan, China
{zhanglin.imre, yanbai, lingyu}@pku.edu.cn, {luoyong, dubo}@whu.edu.cn

Abstract

Federated Learning (FL) aims to establish a shared model across decentralized clients under the privacy-preserving constraint. Despite certain success, it is still challenging for FL to deal with non-IID (non-independent and identical distribution) client data, which is a general scenario in real-world FL tasks. It has been demonstrated that the performance of FL will be reduced greatly under the non-IID scenario, since the discrepant data distributions will induce optimization inconsistency and feature divergence issues. Besides, naively minimizing an aggregate loss function in this scenario may have negative impacts on some clients and thus deteriorate their personal model performance. To address these issues, we propose a Unified Feature learning and Optimization objectives alignment method (FedUFO) for non-IID FL. In particular, an adversary module is proposed to reduce the divergence on feature representation among different clients, and two consensus losses are proposed to reduce the inconsistency on optimization objectives from two perspectives. Extensive experiments demonstrate that our FedUFO can outperform the state-of-the-art approaches, including the competitive one data-sharing method. Besides, FedUFO can enable more reasonable and balanced model performance among different clients.

1. Introduction

Nowadays, the machine learning based artificial intelligence (AI) technologies often rely heavily on large amounts of training data. However, together with the over-collection and over-utilization of personal private data, the risks of privacy disclosure and abuse are increased. For example, in finance, medical treatment and smart city applications, data leakage may lead to huge loss on properties, and even life.

*Ling-Yu Duan is the corresponding author.

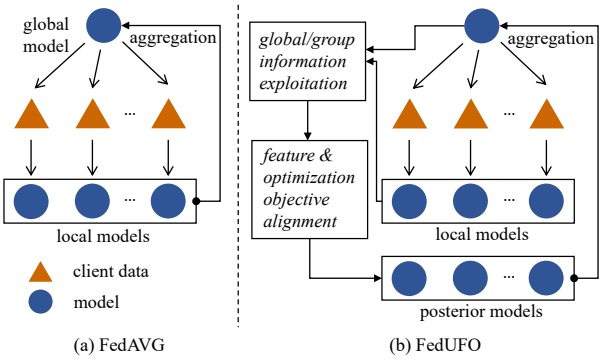

Figure 1. A comparison of the traditional FedAVG and the proposed FedUFO. By making use of the global, group data information extracted from global and local models, our FedUFO effectively aligns the feature representation and optimization objective on local, group and global levels.

To establish health and sustainable AI ecosystem, McMahan *et al.* [20] proposed a new machine learning paradigm - Federated Learning (FL), which breaks the data barriers across different clients (such as regions and industries) under privacy-protection constraint. The main idea of FL is to change the transmission and aggregation from data-level to model-level, so that the advantage of big data can be taken in the AI applications without privacy disclosure. Inspired by this paradigm, data-preserving decentralized learning has been studied in a variety of applications [3, 4, 5, 18].

In real-world FL tasks, data distributions of different clients may vary since the data are usually collected from different sources or scenarios. However, existing FL approaches often simply ignore the distribution divergence, and it has been demonstrated that performance of the global model on non-IID (non-independent and identical distribution) federated data can be much worse than that on the IID ones [22, 9, 16]. The reason is proved to be the weight divergence derived from the discrepant data distribution [32, 19], i.e., the local models are optimized to different directions, which are far from the ideal. The distribution divergence of

non-IID data will also aggravate the performance fairness issue [15], which means that the established model yields significantly different local performance across clients.

This motivates us to develop a novel non-IID FL method by simultaneously aligning the local optimization objectives across clients. Specifically, a group consensus loss (CGR loss) and a global consensus loss (CGL loss) are proposed to explicitly reduce optimization objective inconsistencies by considering a group of clients and all clients respectively. This is achieved by fixing the biased distribution of model predictions by utilizing the extracted group and global data information. Besides, an adversary scheme with a unified adversarial loss (UAD loss) is proposed to implicitly align the optimization objectives on the feature level, and this enforces the model learning generalized feature representation across all data distributions. Fig. 1 is a comparison between the proposed FedUFO and traditional FedAVG [20] methods. Compared with FedAVG, which only utilizes the local data during local training without considering the optimization objective differences, our FedUFO can exploit the additional global and group data information for objective alignment, and hence the obtained models are more reliable.

According to the extensive experiments, it has been verified that simply using the proposed UAD loss can outperform the existing approaches, and performance of our model can be further improved by adding the CGR and CGL losses. Our FedUFO can also achieve more fair performance distribution across clients. Moreover, by evaluating FedUFO on four challenging re-identification and classification tasks, the effectiveness of FedUFO in real-world FL applications is validated.

To summarize, the main contributions of this paper are:

- We design a group consensus loss and a global consensus loss to explicitly align the local, group and global optimization objectives to alleviate the varied data distribution issue in non-IID FL.
- We propose an adversary scheme with a unified adversarial loss to learn unified feature representation that can further help align the optimization objectives across clients implicitly.
- Numerous experiments are conducted in various computer vision tasks. The results demonstrate superiority of the proposed method in real-world applications.

2. Related Work

Federated Learning (FL) [20] aims to train models using decentralized datasets under the data-preserving constraint. Non-IID federated data is general in realistic FL scenarios [9], and the performance will drop significantly when directly applying the traditional FL methods for IID federated data [22, 9, 16]. Hsu *et al.* [7] verify that the performance decrease becomes larger with the increasing of skewness in data distribution. Therefore, some FL approaches are designed for non-IID data, and existing solutions can be roughly grouped as two categories: server-centric and client-centric methods.

Server-centric solution. Such methods aim to build a global model that performs well on all datasets, which are more valuable and useful in real-world applications. Zhao *et al.* [32] quantity the dataset skewness by the earth mover's distance(EMD) and propose a data-sharing strategy to reduce the EMD across clients. Liu *et al.* [16] design an algorithm that computes multiple local aggregations on edges to alleviate the client-edge and edge-server gradient divergence. Saltter *et al.* [21], He *et al.* [6] and Wang *et al.* [26] design gradient compression, neural architecture search and client selection algorithms on non-IID FL scenario to improve the transmission, model architecture and convergence rate respectively. In this paper, we follow the idea of the server-centric method. Specifically, we align the optimization objective across clients with unified feature representation, so that the influence of discrepant data distributions will be alleviated without any data sharing.

Client-centric solution. Such methods do not pursue an effective global model, but attempt to build local model for every client that is only good at its own dataset, which is more feasible in extreme conditions. Li *et al.* [14] construct consensus across clients with a public dataset to restrict the local model training. Shen *et al.* [22] design a new FL paradigm that realizes mutual learning between the global and local models. Wu *et al.* [29] and Zhuang *et al.* [34] split the federated model into a feature extractor and a classifier and use different strategies to transfer the knowledge between the local and global models. These researches treat the data information from other clients as a kind of knowledge and use it to guide the local training. However, building a set of local models is not the target of this paper.

Apart from performance improvement, the proposed method can achieve more fair performance distribution across clients, which is also explored by q-FFL [15]. However, q-FFL performs bad in non-IID scenarios. Besides, it only focuses on fairness, and the overall model performance may be poor. Whereas in our method, both the good overall performance and fairness between different clients are well guaranteed.

3. Methods

In federated learning, the non-IID client data will lead to significant performance drop when disregarding optimization objective difference derived from discrepant data distribution [7, 32], and performance of the model is often much worse than that in the IID data settings [22, 9, 16]. In this paper, we propose the Unified Feature learning and Optimization objective alignment method (FedUFO) for non-IID FL. Specifically, an adversarial module with an uniform loss is proposed to learn unified feature representa-

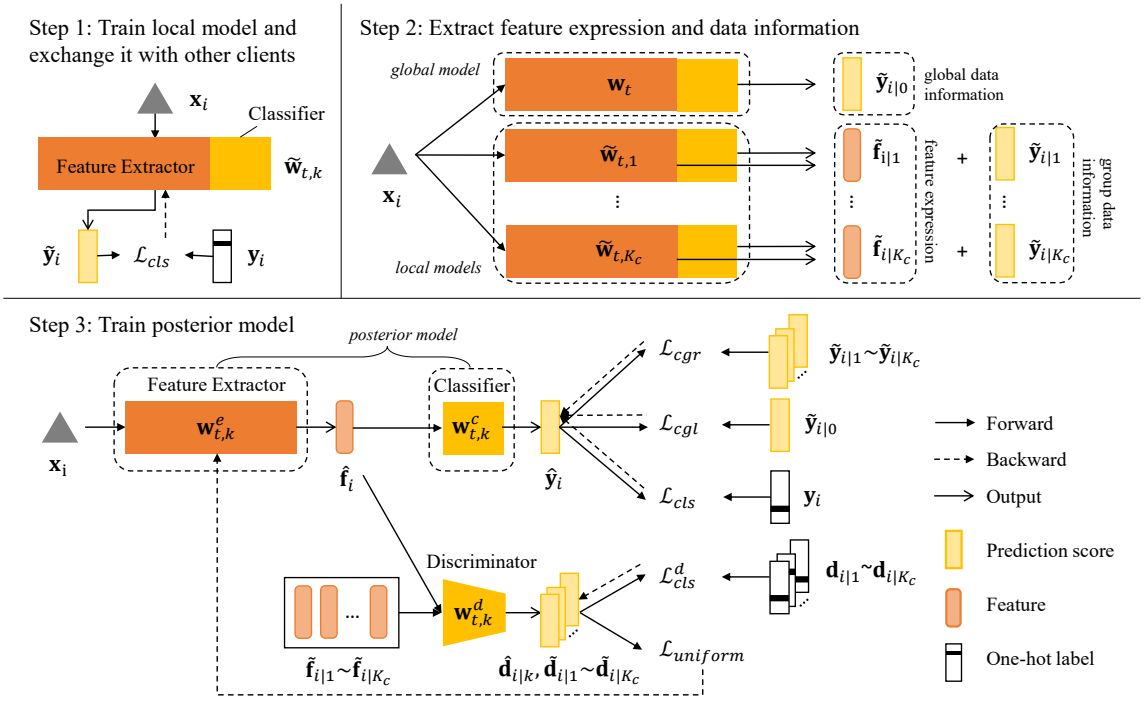

Figure 2. The whole training procedure of k-th client on t-th iteration. Step 1: Train local model $\widetilde{w}_{t,k}$ using the local data and exchange it with other clients. Step 2: Extract the feature representation, global and group data information through the global model w_t and all local models $\widetilde{w}_{t,1} \sim \widetilde{w}_{t,K_c}$. Step 3: Train posterior model $w_{t,k}$ under aligned feature representation and optimization objective constraints.

tion across clients, which implicitly aligns the optimization objective on feature level. And two consensus losses are applied to fix the biased distribution of model prediction considering a group of clients and all clients, which align the optimization objective explicitly.

3.1. Overall Framework

Our FedUFO follows the similar training mechanism of server-centric methods (such as FedAVG [20]): in each iteration, the server sends the global model to *a group of clients*; then these clients train the global model separately with local dataset and upload the generated models to the server; and finally the server updates the global model with a weighted aggregation of the generated models. In this paper, we mainly optimize the client training procedure with an adversary module and two consensus losses, so that the uploaded models share the same optimization objective.

Fig. 2 illustrates the training procedure (three training steps) of the k-th client in the t-th iteration: 1) based on the global model $\mathbf{w}_t$, a local model $\widetilde{\mathbf{w}}_{t,k}$ is trained with local dataset $\mathcal{D}_k$ only, which contains data information and feature representation of the k-th client. $\widetilde{\mathbf{w}}_{t,k}$ can be regarded as a prior model, and the client group exchange the local models with each other; 2) the k-th client extracts the global data information from the global model, and also extracts group data information and feature representation from all local models. The extracted global and group data information are used to establish global and group optimization objectives respectively; 3) a posterior model $\mathbf{w}_{t,k}$ is trained locally under the constraints of unified feature representation and optimization objective, and finally sent to the server for model aggregation and update.

In the following, we use K to denote the number of clients. $K_c = C * K$ is the size of the client group [20], and C is the fraction ($C \in [0, 1]$). S_t is the group in the t-th iteration ($|S_t| = K_c$). Dataset of the k-th client is defined as $\mathcal{D}_k = \{(\mathbf{x}_{i,k}, \mathbf{y}_{i,k}) | 1 \leqslant i \leqslant n_k\}$, where n_k is the dataset size. A model h consists of two parts - a feature extractor h_e and a classifier h_c. $\mathbf{w}_t$ is the global model parameter in the t-th iteration. $\widetilde{\mathbf{w}}_{t,k}$ and $\mathbf{w}_{t,k}$ are the local and posterior model parameter respectively. $\hat{\mathbf{f}}_{i,k}$ and $\hat{\mathbf{y}}_{i,k}$ are the output feature and prediction of the sample $\mathbf{x}_{i,k}$ by the posterior model $\mathbf{w}_{t,k}$. $\widetilde{\mathbf{f}}_{i,k|j}$ is feature of $\mathbf{x}_{i,k}$ upon the local model of the j-th client $\widetilde{\mathbf{w}}_{t,j}$, and $\widetilde{\mathbf{y}}_{i,k|j}$ is the corresponding prediction. For notation simplicity, we ignore the subscript k in the following. $\mathbf{d}_{i|k}$ and $\mathbf{d}_{i|j}$ represent the one-hot ground-truth labels of features $\hat{\mathbf{f}}_i$ (i.e., $\mathbf{x}_{i,k}$) and $\widetilde{\mathbf{f}}_{i|j}$.

3.2. Unified Feature Representation Learning

The different data distributions will lead to different data feature the local models focus on. Consequently, the clients will generate disparate feature representation, which will hinder optimization objective alignment on feature level.

Thus, we first align the feature representation across clients to implicitly unify the optimization objectives. Specifically, we introduce adversary learning together with a feature discriminator D upon the feature extractor h_e. During the training procedure, D is learned to recognize which client the input feature belongs to, while h_e tries to learn distribution-free feature representation across clients.

After obtaining the local models of all clients in S_t except itself, we first compute the data features of $\mathbf{x}_i$ on these local models,

$$\widetilde{\mathbf{f}}_{i|j} = h_e(\mathbf{x}_i|\widetilde{\mathbf{w}}^e_{t,j}), 1 \leqslant j \leqslant K_c, j \neq k, \quad (1)$$

where $\widetilde{\mathbf{w}}^e_{t,j}$ is the parameter of feature extractor of j-th local model $\widetilde{\mathbf{w}}_{t,j}$. Here $\widetilde{\mathbf{f}}_{i|j}$ implies the interested feature of j-th client, i.e., the general feature of j-th data distribution. During the training of the current k-th posterior model, features from the local models $\{\widetilde{\mathbf{f}}_{i|j}\}_{j\neq k}$ and posterior model $\hat{\mathbf{f}}_i$ will be fed into the discriminator to predict which client (feature representation) they belong to,

$$\widetilde{\mathbf{d}}_{i|j} = Softmax(D(\widetilde{\mathbf{f}}_{i|j}|\mathbf{w}^d_{t,k})), 1 \leqslant j \leqslant K_c, j \neq k, \quad (2)$$

$$\hat{\mathbf{d}}_i = Softmax(D(\hat{\mathbf{f}}_i|\mathbf{w}^d_{t,k})), \quad (3)$$

where $\mathbf{w}^d_{t,k}$ stands for the parameter of D. The $\widetilde{\mathbf{d}}_{i|j}$ and $\hat{\mathbf{d}}_i$ are K-dimensional vectors.

To enforce the extractor h_e to learn unified feature representation, feature $\hat{\mathbf{f}}_i$ is expected to have the same prediction scores across all classes. Thus, we use a uniform adversarial loss (UAD loss) to constrain the prediction score $\hat{\mathbf{d}}_i$,

$$\mathcal{L}_{uniform} = -\frac{1}{K}\sum_{j=1}^{K} log(\hat{\mathbf{d}}^j_i), \quad (4)$$

where $\hat{\mathbf{d}}^j_i$ is the j-th element of $\hat{\mathbf{d}}_i$. Note that $\mathcal{L}_{uniform}$ is smallest when $\hat{\mathbf{d}}^j_i = \frac{1}{K}, \forall j \in [1, K]$.

To enable the discriminator distinguishing features extracted using different local models, all the prediction results $\widetilde{\mathbf{d}}_{i|j}$ and $\hat{\mathbf{d}}_i$ are used to optimize the discriminator. We use the client index l_k to denote the class of the feature representation of the k-th client, and by adopting cross-entropy loss, we have the following total classification loss:

$$\mathcal{L}^d_{cls} = -log(\hat{\mathbf{d}}^{l_k}_i) - \frac{1}{K_c - 1}\sum_{j=1, j\neq k}^{K_c} log(\widetilde{\mathbf{d}}^{l_j}_{i|j}) \quad (5)$$

3.3. Unified Optimization Objective Alignment

There are two types of optimization objective inconsistency in non-IID FL: the local-group inconsistency and group-global inconsistency. Here, the local, group and global optimization objectives correspond to one client, client group S_t and all clients. The local-group optimization objective inconsistency will induce model exclusion during aggregation. The group-global optimization objective inconsistency will not only make the global model update unsmooth, but also deteriorate the local model performance of clients out of S_t. Therefore, these two inconsistencies will result in low overall model performance [7] and unfair performance distribution [15]. To improve the performance and performance fairness, we propose to reduce the inconsistency by constructing consensus on the local, group and global levels. Specifically, a group consensus loss (CGR loss) and a global consensus (CGL loss) loss are proposed to align the local-group and group-global optimization objectives respectively.

3.3.1 Group Consensus Loss

We use the CGR loss to construct consensus among the client group S_t in each iteration to reduce the local-group optimization objective inconsistency. The local optimization objective is directly reflected from the model parameter changes between local and global models. However, the model parameters are highly information concentrated and of large-scale, thus are difficult to compare and merge. To tackle this issue, we use distributions of the predictions for the same data on different local models to characterize different local optimization objectives. Then we establish the group optimization objective with all local optimization objectives in group S_t.

We first calculate the predictions for $\mathbf{x}_i$ by local models of all the other clients:

$$\widetilde{\mathbf{y}}_{i|j} = Softmax(h(\mathbf{x}_i|\widetilde{\mathbf{w}}_{t,j})), 1 \leqslant j \leqslant K_c. \quad (6)$$

where $\widetilde{\mathbf{y}}_{i|j}$ is the prediction of $\mathbf{x}_i$ using the local model $\widetilde{\mathbf{w}}_{t,j}$. Here, $\widetilde{\mathbf{w}}_{t,j}$ is the adaptation of global model on the j-th data $\mathcal{D}_j$, thus it implies the optimization direction of the j-th client, and so does $\widetilde{\mathbf{y}}_{i|j}$. To align the optimization objectives across clients in S_t, we merge all the $\widetilde{\mathbf{y}}_{i|j}$ considering the category data proportion to generate a unified prediction:

$$\widetilde{\mathbf{y}}_i = Softmax(\sum_{j=1}^{K_c} \mathbf{p}_j \otimes \widetilde{\mathbf{y}}_{i|j}), \quad (7)$$

where $\otimes$ is the element-wise product, $\mathbf{p}_j$ is the data proportion of every category of $\mathcal{D}_j$ against the total category data number in the group; $\widetilde{\mathbf{y}}_i$ provides an expected prediction after considering all the client data information in S_t. We regard $\widetilde{\mathbf{y}}_i$ as an aligned optimization objective and use it to restrict the prediction distribution of $\mathbf{w}_{t,k}$.

To enable the local optimization objective approaching the group optimization objective, we restrict the prediction of posterior model $\hat{\mathbf{y}}_i$ to have the same class probability distribution as $\widetilde{\mathbf{y}}_i$ in Eq. 7. The Kullback-Leibler divergence is adopted as the loss function, which can capture the difference of two probability distributions, thus the CGR loss is given by

$$\mathcal{L}_{cgr} = KLDiv(\hat{\mathbf{y}}_i||\widetilde{\mathbf{y}}_i), \quad (8)$$

3.3.2 Global Consensus Loss

The CGL loss is proposed to align the group and global optimization objectives. The group optimization objective is

defined by using the total data of client group. Due to the prohibition of data aggregation, it is impossible to restrict the group optimization objective directly. To reduce the group-global objective inconsistency, we restrict the local model update with respect to the global data information, and thus the group optimization objective will be adjusted according to the global optimization objective. This can be achieved by using the same strategy as in the CGR loss, i.e., extracting the prediction from all the K client models and generating a unified prediction distribution. However, this will result in large computation and transmission costs. Instead, we use the aggregated global model of the last iteration (i.e., $\mathbf{w}_t$) to construct the global consensus, which will accumulate the global data information with continuous training.

Similarly, we first receive the prediction $\widetilde{\mathbf{y}}_{i|0}$ of $\mathbf{x}_i$ on the global model,

$$\widetilde{\mathbf{y}}_{i|0} = Softmax(h(\mathbf{x}_i|\mathbf{w}_t)), \tag{9}$$

where $\widetilde{\mathbf{y}}_{i|0}$ has the same effect as $\widetilde{\mathbf{y}}_i$ in Eq. 7, i.e., provides an expected prediction distribution after considering all data information. Then we compute CGL loss with prediction $\hat{\mathbf{y}}_i$ from posterior model to align the local and global optimization objectives:

$$\mathcal{L}_{cgl} = KLDiv(\hat{\mathbf{y}}_i||\widetilde{\mathbf{y}}_{i|0}). \tag{10}$$

Dynamic Sampling Strategy. The alignment of group and global optimization objectives can alleviate the performance drop of the clients not included in the current client group. To further achieve a fair performance distribution, a dynamic sampling strategy is adopted together with CGL loss, where $\frac{K_c}{2}$ clients that have the lowest local performance are selected and form the group S_t with other random $\frac{K_c}{2}$ clients. Compared with the existing methods that select K_c clients randomly, this strategy can help improve the performance fairness.

3.4. Training Procedure

The whole training procedure consists of two stages.

Stage One: Train federated model and discriminator with constraint on feature level. In each iteration, we first freeze the parameter of the discriminator D and train the federated model h with the following loss,

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \mathcal{L}_{uniform}, \tag{11}$$

where $\mathcal{L}_{cls}$ is the cross-entropy loss. Then parameters of the federated model are fixed and we train the discriminator using $\mathcal{L}^d_{cls}$ in Eq. 5. After that, unified feature representation that is effective on all client data can be obtained.

Stage Two: Fine-tune federated model with both feature and objective consistency constraints. Based on the federated model and discriminator from stage one, we fine-tune the federated model under the CGR loss, the CGL loss and the UAD loss. Specifically, the $\mathcal{L}_{total}$ in Eq. 11 becomes

Method	MNIST	CIFAR10
centralized (upper bound)	*97.89±0.08*	*81.91±0.47*
FedAVG [20]	94.38±1.72	59.95±3.08
FedMeta [31]	94.23±0.41	66.29±0.34
FML [22]	93.73±1.39	51.72±4.18
FedRetile [5]	95.00±0.10	63.40±0.54
q-FFL [15]	87.70±1.39	46.25±4.10
FedUAD(ours)	96.70±0.20	65.29±2.07
FedUFO(ours)	**96.75±0.07**	**67.45±0.81**

Table 1. Model performance.

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \mathcal{L}_{uniform} + \mathcal{L}_{cgr} + \lambda\mathcal{L}_{cgl}, \tag{12}$$

where λ is a trade-off hyper-parameter.

4. Experiments

In this Section, we conduct extensive experiments to validate the effectiveness of FedUFO, including four real-world re-identification and classification tasks. In Section. 4.1, we compare the model performance and performance fairness of FedUFO with existing methods. Section. 4.2 is the ablation study about all the proposed modules and the hyper-parameter λ. In Section. 4.3, we test FedUFO on more challenging real-world scenarios.

4.1. Performance Comparison

4.1.1 Settings

i. Dataset. Following [20], MNIST [13] and CIFAR-10 [12] are used to evaluate the methods. Each dataset is partitioned into 100 clients: data of each category is divided into 20 shards, and each client is randomly assigned 2 shards from the total 200 shards (10 classes).

ii. Implementation. We use the same network architecture as in [20] for MNIST and CIFAR10. For MNIST, the batch-size B = 10, the client fraction C = 0.1 and the local training epoch E = 10, while B = 50 for CIFAR-10 instead. For both two datasets, the discriminator of two fc layers and ReLU activation is used, where the output size of the first layer is the same as the input (200 and 64 respectively). We use the SGD optimizer in both client and server, with 0.01 as learning rate, 0.9 as momentum and 0.0002 as weight decay. λ in Eq. 12 is set to be 3 on both of the datasets.

iii. Evaluation Metrics. The classification accuracy P_{acc} is used to evaluate the model performance. Similar as [15], the best, worst local performance and the stand deviation(std) of all the local performances are used to evaluate the performance fairness, which are termed as P_b, P_w and P_{std}.

iv. Comparison Methods. We compare FedUFO with FedAVG [20], FedMeta [31], FML [22], FedRetile [5] and q-FFL [15]. They adopt the same implementation settings as in **ii. Implementation** for fair comparison. Apart from the common settings, for FedMeta, 10% data of every client are

		centralized (upper bound)	comparison methods					ours	
			FedAVG	FedMeta	FML	FedRetile	q-FFL	**FedUAD**	**FedUFO**
MNIST	P_b	*100*	98.95	98.86	98.75	**100**	98.10	**100**	**100**
	P_w	*91.55*	86.80	85.00	85.88	82.40	71.81	**89.69**	**89.69**
	P_{std}	*1.47*	2.81	2.80	2.84	3.79	4.99	2.05	**1.87**
CIFAR10	P_b	*92.30*	73.60	81.30	70.80	**83.90**	62.90	78.60	75.40
	P_w	*66.40*	32.50	46.00	36.00	30.10	26.20	47.80	**55.40**
	P_{std}	*5.21*	9.23	7.63	8.02	12.77	8.16	7.07	**3.54**

Table 2. Performance fairness comparison on MNIST and CIFAR-10.

used to construct the meta data that finetune the aggregated global model in each iteration. For FML, performances of the global models are reported for fair comparison. For q-FFL, we test $q = 1, 2, 5$ and provide the result of $q = 2$, which is also the best result.

4.1.2 Model Performance

Table. 1 displays the model performance of different methods on MNIST and CIFAR10 datasets. The centralized learning is provided as "*centralized*" in the table, which is the upper bound of the FL algorithms. The FedUAD and FedUFO represent the model trained after *stage one* and *stage two* respectively. From the table, we can see that only applying the adversarial scheme already achieves higher performance than non-data-sharing methods. This verifies that the proposed adversarial module helps the federated model learning unified feature representation, so that the optimization objective divergence can be alleviated on feature level. After appending the consensus losses, the model performance further improves, even exceeds the FedMeta on both datasets, where the clients share a fraction of data with the server. q-FFL is proposed to solve the unfair local performance distribution. We can find that the emphasis of q-FFL on performance fairness is at the expense of model performance, while our FedUFO can maintain and even improve the model performance. This is because under the unified optimization objective, the model exclusion will be greatly mitigated, and performances on all clients will improve simultaneously in FedUFO.

4.1.3 Performance Fairness

Table. 2 compares P_b, P_w and P_{std} of all the methods. We highlight the highest P_b and P_w as well as the lowest P_{std} second to the results of "*centralized*" in bold. On both datasets, FedUFO yields the lowest P_{std} among all compared methods, even exceeds the centralized learning on CIFAR-10, which means the proposed adversary scheme and consensus losses can result in a fairer performance distribution. FedUAD has the second lowest P_{std}, as the unified feature representation can also alleviate the mutual exclusion among local models on feature level. This can also be verified by the results on P_w, where FedUFO and FedUAD are the first and second highest respectively. FedRetile has higher P_b than FedUAD and FedUFO on CIFAR-10. However, it yields the highest P_{std} in the meantime. This reveals the character of some FL algorithms: enhancing the model performance by only improving the local performance of the easy-to-learned clients, while the other clients are out of the consideration. q-FFL performs badly in non-IID FL scenario. As the motivation of q-FFL is to emphasize the clients with lower local performance, it can not tackle optimization divergence in non-IID scenario, thus overemphasis of one client will lead to the performance drop on other clients.

Method		P_{acc}
baseline	`cls`	59.95±3.08
one module	`+cgr`	65.10±1.02
	`+cgl`	65.01±3.45
	`+uad`	65.29±2.07
two modules	`+uad+cgr`	66.29±1.59
	`+uad+cgl`	66.30±1.63
all modules	`+all`	67.80±0.87

Table 3. Model performance comparison across different variations of FedUFO on CIFAR10.

4.2. Ablation Study

In this section, we verify the effectiveness of proposed modules by splitting and combining the UAD loss, CGR loss and CGL loss one by one. `cgr`, `cgl` and `uad` represent applying the CGR loss, CGL loss and UAD loss respectively. `all` is the model trained with all three losses. The baseline method that only uses classification loss is denoted as `cls`, but it equals to FedAVG method actually and we just copy the results from Table. 1. For a fair comparison, the implementation settings are the same on all ablation methods (Section. 4.1.1).

4.2.1 Model Performance

Table. 3 displays the model performance of different variations of FedUFO. `cgr` and `cgl` yield lower performance than `uad` in the table. This indicates that only applying the consensus losses without unified feature representation can not align the optimization objectives across clients. After

		cls	one module			two modules		+all
			+cgr	+cgl	+uad	+uad+cgr	+uad+cgl	
MNIST	P_b	98.95	98.79	99.52	**100**	**100**	**100**	**100**
	P_w	86.80	87.01	88.87	89.69	89.28	**89.79**	89.18
	P_{std}	2.81	2.39	2.06	2.05	1.95	**1.90**	1.98
CIFAR10	P_b	73.60	78.50	**83.10**	78.60	81.10	75.00	80.60
	P_w	32.50	43.80	43.30	47.80	45.50	**55.80**	49.30
	P_{std}	9.23	7.87	8.56	7.07	7.63	**4.21**	7.42

Table 4. Performance fairness comparison across different variations of FedUFO on MNIST and CIFAR-10.

Method	P_{acc}	P_{std}
cgl(w/)	63.07±2.93	3.65
cgl(w/o)	65.01±3.45	8.56
uad+cgl(w/)	66.56±0.81	3.53
uad+cgl(w/o)	66.30±1.63	4.21
all(w/)	67.45±0.81	3.54
all(w/o)	67.80±0.87	7.42

Table 5. Performance and fairness comparison with (w/) and without (w/o) dynamic sampling strategy on CIFAR10.

the training of discriminator, the uad+cgr and uad+cgl yields better performance than cgr and cgl, and they improve the performance of uad as well. Besides, we notice that the addition of any of the UAD, CGR and CGL losses will lead to performance improvement.

FedUFO with CGL loss and CGR loss always achieve comparable performance in Table. 3, while CGR loss brings in extra transmission cost than CGL loss and FedAVG algorithm. In practice, if the network bandwidth is limited, only using the CGL loss can produce a decent model as well.

4.2.2 Performance Fairness

Table. 4 compares the performance fairness among different variations of FedUFO. From this table, we can draw the following conclusions.

1) The performance fairness can be improved by every module and their combinations.

2) Among three proposed losses, UAD loss achieves the fairest performance distribution. Besides, after combining with UAD loss, uad+cgr and uad+cgl yields lower P_{std} than before (cgr and cgl). These verify the importance of feature-level alignment across clients.

3) uad+cgl achieves the best P_{std} and P_w. This verifies that CGL loss improves the performance of clients in group without sacrificing others.

4) Appending cgr may hurt the fairness, as all yields higher P_{std} than uad+cgl on both datasets.

4.2.3 Effect of Dynamic Sampling Strategy

Table. 5 compares the performance and fairness with and without dynamic sampling strategy on CIFAR10. In Table. 5, dynamic sampling strategy yields lower performance than without it, but it improves the fairness inversely. This indicates that the emphasis on fairness will damage the model performance, as the training will put more on local model training with low local performance. In practice, a fair FL model that achieves similar performance on all clients will motivate more data providers join the federation, which further facilitates the development of FL.

4.2.4 Convergence Rate

Fig. 3 illustrates the model performance changes during the first 100 iterations on MNIST. We can find that the methods with CGL loss(cgl, uad+cgl and all) converges slower than the other methods. cgr, uad and uad+cgr have similar performance trends as FedAVG, which means applying UAD loss and CGR loss will not influence the original convergence rate.

Fig. 4 displays the effect of λ on the convergence rate. When $\lambda > 3$, the model converges quite slow, even be constant. λ=1, 2 and 3 yield a similar convergence rate. In our experiments, $\lambda = 3$ achieves the best performance.

Note that the learning curve in Fig. 3 and Fig. 4 are after fitted for better illustration. And the cgl are used together with dynamic sampling strategy.

4.3. Experiments on More Challenging Scenarios

In this section, we test FedUFO on more challenging real-world tasks - vehicle reID (VeRI776[17]), person reID (MSMT17[33] and Market1501[27]) and fine-grained classification (CUB200[25]), which are also important applications of FL. Similar as in Section. 4.1.1, each dataset is partitioned into 100 clients. We adopt Resnet50 as the network architecture and three fc layer with ReLU activation as discriminator. The other settings are the same as CIFAR10 in Section. 4.1.1. Note that for datasets VeRI776, MSMT17 and Market1501, the model performance is evaluated with *mean average precision* (mAP, denoted as P_{map}), and the feature is L2-normalized before retrieval. The experiment results are illustrated in Table. 6.

From the table, we find that FedUAD and FedUFO consistently outperform the other methods on model performance (P_{map} or P_{acc}), which verifies the effects of the proposed method on real-world FL applications. FedRetile is

	VeRI776 [17]		MSMT17 [27]		Market1501 [33]		CUB200 [25]	
	P_{map}	P_{std}	P_{map}	P_{std}	P_{map}	P_{std}	P_{acc}	P_{std}
centralized	*65.34±0.12*	*1.75e-3*	*34.20±0.26*	*1.23e-3*	*57.91±1.93*	*2.72e-3*	*89.49±0.21*	*1.22e-1*
FedAVG [20]	58.51±0.89	6.13e-3	25.83±1.45	1.93e-3	49.82±1.80	1.31e-1	55.12±1.41	1.58e-1
FedRetile [5]	59.69±0.14	4.36e-3	29.07±0.13	1.87e-3	50.16±0.13	1.26e-1	55.74±0.34	1.43e-1
FML [22]	58.85±1.94	5.72e-3	26.56±0.68	1.96e-3	49.90±1.51	5.00e-2	46.65±2.42	1.85e-1
FedUAD(ours)	60.29±0.15	1.95e-3	29.98±0.78	1.87e-3	51.05±0.58	5.97e-2	59.20±1.33	1.61e-1
FedUFO(ours)	**60.91±0.20**	**1.92e-3**	**30.22±0.16**	**1.77e-3**	**51.68±0.42**	**4.65e-2**	**60.57±0.53**	**1.36e-1**

Table 6. Performance (P_{map} or P_{acc}) and fairness (P_{std}) comparison on VeRI776, MSMT17, Market1501, CUB200.

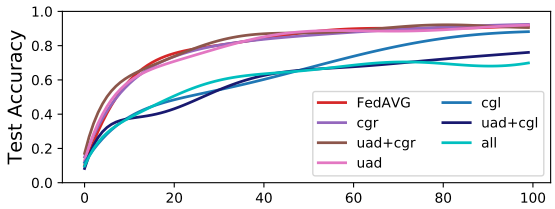

Figure 3. Model performance changes of different methods in the first 100 iterations on MNIST.

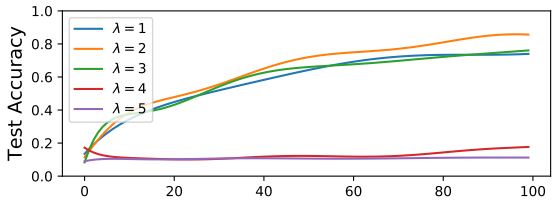

Figure 4. Model performance changes of `cgl` with different λ in the first 100 iterations on MNIST.

inferior to FedUAD and FedUFO in all cases, as it only changes the weights of models during aggregation, but does not alleviate the model divergence derived from different data distributions. Besides, FedUFO yields the lowest P_{std} on four datasets, which means the established model achieves the fairest performance across different datasets. P_{std} of FedUAD exceeds the counterparts in some datasets. This is because UAD loss does not directly act on the optimization objective, but on the feature representation, thus its effect on reducing performance unfairness is limited. Comparing the centralized training ("centralized") with other methods in Table. 1 and Table. 6, we notice that the data heterogeneity seems to influence more on classification tasks than re-ID tasks. This may be due to the discrepancy of training target (classification error) and test target (retrieval performance) in re-ID. Moreover, FedUFO in classification tasks achieves more performance improvement than re-ID tasks, of which reason is not clarified currently.

5. Discussion

Application scope. The proposed FedUFO contains three modules - UAD loss, CGR loss, CGL loss. Among them, UAD loss and CGR loss are accomplished with additional communication cost and/or storage. Considering the two FL applications proposed by Kariouz et al. [10] FedUFO is more applicable to the cross-silo FL, where the clients are 2∼100 organizations with sufficient communication and storage resources. For the cross-device FL, the clients are massive mobile or IoT devices, thus communication and storage are often the bottlenecks. In this scenario, the proposed CGL loss can be adopted solely with no extra communication cost and little extra storage. Comparing Table. 3 and Table. 4, we can find that CGL loss already exceeds the baselines.

Model Extension. The proposed FedUFO explores a new FL model training scheme, which is especially effective in non-IID FL scenarios. Apart from the model training, the establishment of a complete FL framework requires a series of auxiliary mechanisms, such as incentive mechanism [28], model compression [11, 2], backdoor defense [24, 30, 23], communication protocol [1, 8], etc., which will guarantee the healthy development of FL ecosystem. We have found that FedUFO can be seamlessly embedded into most of the existing methods on these topics, as FedUFO only changes the local model training procedure and maintains the overall training pipeline.

6. Conclusion

In this paper, we analyze the reason for the low model performance and unfair performance distribution under the non-IID federated learning (FL) scenario, and propose a novel FedUFO algorithm that simultaneously aligns the feature representation and optimization objective across clients under the data-preserving constraint. For the feature representation, an unified adversarial loss is designed to enable the model learning cross-client features. In regard to the optimization objective, two consensus losses are designed to mitigate the optimization inconsistency on local, group and global levels. Extensive experiments on various computation vision tasks demonstrate effectiveness of the proposed FedUFO compared with the state-of-the-art FL approaches. In the future, we intend to design more sophisticated strategies to align the group and global optimization objectives.

Acknowledgement: This work was supported by the National Natural Science Foundation of China under Grant 62088102, and in part by the PKU-NTU Joint Research Institute (JRI) sponsored by a donation from the Ng Teng Fong Charitable Foundation.

References

[1] Keith Bonawitz, Hubert Eichner, et al. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1:374–388, 2019. 8

[2] Sebastian Caldas, Jakub Konecný, H. Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018. 8

[3] Dashan Gao, Ce Ju, et al. Hhhfl: Hierarchical heterogeneous horizontal federated learning for electroencephalography. In *Adv. Neural Inform. Process. Syst.*, 2019. 1

[4] Andrew Hard, Kanishka Rao, et al. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018. 1

[5] Florian Hartmann, Sunah Suh, Arkadiusz Komarzewski, Tim D Smith, and Ilana Segall. Federated learning for ranking browser history suggestions. In *Adv. Neural Inform. Process. Syst.*, 2019. 1, 5, 8

[6] Chaoyang He, Haishan Ye, et al. Milenas: Efficient neural architecture search via mixed-level reformulation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[7] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. In *Adv. Neural Inform. Process. Syst.*, 2019. 2, 4

[8] Martin Isaksson and Karl Norrman. Secure federated learning in 5g mobile networks. In *2020 IEEE Global Communications Conference*, pages 1–6, 2020. 8

[9] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, et al. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018. 1, 2

[10] Peter Kairouz, H. Brendan McMahan, Brendan Avent, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1), 2021. 8

[11] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, et al. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. 8

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[13] Yann LeCun, Léon Bottou, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[14] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019. 2

[15] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *Int. Conf. Learn. Represent.*, 2019. 2, 4, 5

[16] Lumin Liu, Jun Zhang, S. H. Song, and Khaled Ben Letaief. Edge-assisted hierarchical federated learning with non-iid data. *CoRR*, abs/1905.06641, 2019. 1, 2

[17] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Trans. Multimedia*, 20(3):645–658, 2018. 7, 8

[18] Songtao Lu, Yawen Zhang, et al. Learn electronic health records by fully decentralized federated learning. *arXiv preprint arXiv:1912.01792*, 2019. 1

[19] Grigory Malinovsky, Dmitry Kovalev, Elnur Gasanov, Laurent Condat, and Peter Richtarik. From local sgd to local fixed point methods for federated learning. *arXiv preprint arXiv:2004.01442*, 2020. 1

[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 3, 5, 8

[21] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 2019. 2

[22] Tao Shen, J. Zhang, Xinkang Jia, Fengda Zhang, et al. Federated mutual learning. *arXiv preprint arXiv:2006.16765*, 2020. 1, 2, 5, 8

[23] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, and Ji Liu. Data poisoning attacks on federated machine learning. *arXiv preprint arXiv:2004.10020*, 2020. 8

[24] Ananda Theertha Suresh, Brendan McMahan, Peter Kairouz, and Ziteng Sun. Can you really backdoor federated learning. *arXiv preprint arXiv:1911.07963*, 2019. 8

[25] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 7, 8

[26] H. Wang, Z. Kaplan, D. Niu, and B. Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE Conference on Computer Communications*, 2020. 2

[27] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 79–88, 2018. 7, 8

[28] Shuyue Wei, Yongxin Tong, Zimu Zhou, and Tianshu Song. Efficient and fair data valuation for horizontal federated learning. *Federated Learning*, pages 139–152, 2020. 8

[29] Guile Wu and Shaogang Gong. Decentralised learning from independent multi-domain labels for person re-identification, 2020. 2

[30] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *Int. Conf. Learn. Represent.*, 2020. 8

[31] Xin Yao, Tianchi Huang, Rui-Xiao Zhang, Ruiyu Li, and Lifeng Sun. Federated learning with unbiased gradient aggregation and controllable meta updating. In *Adv. Neural Inform. Process. Syst.*, 2019. 5

[32] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, et al. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. 1, 2

[33] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, et al. Person re-identification meets image search. *arXiv preprint arXiv:1502.02171*, 2015. 7, 8

[34] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, et al. Performance optimization of federated person re-identification via benchmark analysis. In *ACM Int. Conf. Multimedia*, 2020. 2