Personalized Federated Learning with Adaptive Batchnorm for Healthcare

Wang Lu, Jindong Wang, Yiqiang Chen, *Senior Member, IEEE,* Xin Qin, Renjun Xu, Dimitrios Dimitriadis, *Senior Member, IEEE,* and Tao Qin, *Senior Member, IEEE*

Abstract—There is a growing interest in applying machine learning techniques to healthcare. Recently, federated learning (FL) is gaining popularity since it allows researchers to train powerful models without compromising data privacy and security. However, the performance of existing FL approaches often deteriorates when encountering non-iid situations where there exist distribution gaps among clients, and few previous efforts focus on personalization in healthcare. In this article, we propose FedAP to tackle domain shifts and then obtain personalized models for local clients. FedAP learns the similarity between clients based on the statistics of the batch normalization layers while preserving the specificity of each client with different local batch normalization. Comprehensive experiments on five healthcare benchmarks demonstrate that FedAP achieves better accuracy compared to state-of-the-art methods (e.g., **10**%+ accuracy improvement for PAMAP2) with faster convergence speed.

Index Terms—Distributed Computing, Federated Learning, Personalization, Batch Normalization, Healthcare.

1 Introduction

Machine learning has been widely adopted in many applications in people's daily life [1], [2], [3]. Specifically for healthcare, researchers can build models to predict health status by leveraging health-related data, such as activity sensors [4], images [5], and other health information [6], [7], [8]. To achieve satisfying performance, machine learning healthcare applications often require sufficient client data for model training. However, with the increasing awareness of privacy and security, more governments and organizations enforce the protection of personal data via different regulations [9], [10]. In this situation, federated learning (FL) [11] emerges to build powerful machine learning models with data privacy well-protected.

Personalization is important in healthcare applications since different individuals, hospitals or countries usually have different demographics, lifestyles, and other health-related characteristics [12], i.e., the non-iid issue (not identically and independently distributed). Therefore, we are more interested in achieving better personalized healthcare, i.e., building FL models for each client to preserve their specific information while harnessing their commonalities. As shown in Fig. 1, there are three different clients A, B, and C with different statistics of data distributions (e.g., the adult A and the child B may have different lifestyles and activity patterns). Even if federated learning can perform in the standard way, the non-iid issue cannot be easily handled. This will severely limit the performance of existing federated learning algorithms.

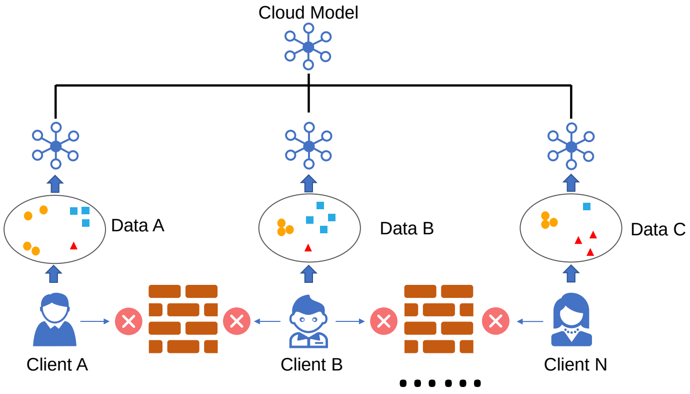


Fig. 1. Data non-iid in federated learning: different clients have different data distributions.

The popular FL algorithm, FedAvg [13], has demonstrated superior performance in many situations [14], [15]. However, FedAvg is unable to deal with non-iid data among different clients since it directly averages the parameters of models coming from all participating clients [16]. There are some algorithms for this non-iid situation. FedProx [17] is designed for non-iid data. However, FedProx only learns a global model for all clients, which means that it is unable to obtain personalized models for clients. FedHealth [18], another work for personalized healthcare, needs access to a large public dataset, which is often impossible in real applications. FedBN [19] handles the non-iid issue by learning local batch normalization layers for each client but ignores the similarities across clients that can be used to boost the personalization.

In this article, we propose FedAP, a personalized feder-

- *Wang Lu and Xin Qin are with University of Chinese Academy of Sciences and Institute of Computing Technology, Chinese Academy of Sciences. E-mail: {luwang, qinxin18b}@ict.ac.cn*
- *Jindong Wang and Tao Qin are with Microsoft Research Asia. E-mail: {jindong.wang, taoqin}@microsoft.com*
- *Yiqiang Chen is with Institute of Computing Technology (CAS) and Pengcheng Laboratory. E-mail: yqchen@ict.ac.cn*
- *Renjun Xu is with Zhejiang University. E-mail: rux@zju.edu.cn*
- *Dimitrios Dimitriadis is with Microsoft Research. E-mail: didimit@microsoft.com*
- *Correspondence to Jindong Wang and Yiqiang Chen.*

Manuscript received May 19, 2022.

ated learning algorithm via *adaptive batch normalization* for healthcare. Specifically, FedAP learns the similarities among clients with the help of a pre-trained model that is easy to obtain. The similarities are determined by the distances of the data distributions, which can be calculated via the statistical values of the layers' outputs of the pre-trained network. After obtaining the similarities, the server averages the models' parameters in a personalized manner and generates a unique model for each client. Each client preserves its own batch normalization and updates the model with a momentum method. In this way, FedAP can cope with the non-iid issue in federated learning. FedAP is extensible and can be deployed to many healthcare applications.

Our contributions are as follows:

1) We propose FedAP, a personalized federated learning algorithm via adaptive batch normalization for healthcare, which can aggregate the information from different clients without compromising privacy and security, and learn personalized models for each client.
2) We evaluate the performance of FedAP in five public healthcare datasets across time series and image modalities. Experiments demonstrate that our FedAP achieves significantly better performance than state-of-the-art methods in all datasets.
3) FedAP reduces the number of rounds and speeds up the convergence to some extent. Moreover, some experimental results illustrate FedAP may be able to reduce communication costs with little performance degradation via increasing local iterations and decreasing global communications.

2 Related Work

2.1 Machine Learning and Healthcare

With the rapid development of perception and computing technology, people can make use of machine learning to help doctors diagnose [20] and assist doctors in the operation [21], etc. Many methods are proposed to monitor people's health state [22] and diagnose diseases that may even have better performance than doctors', especially in the field of medical images [23]. Moreover, machine learning can make disease warnings via daily behavior supervision with simple wearable sensors [24]. For instance, certain activities in daily life reflect early signals of some cognitive diseases. Through daily observation of gait changes and finger flexibility, the machine can tell people whether they are suffering from Parkinson [25]. In addition, some studies worked for better personalization in healthcare [26], [27].

Unfortunately, a successful healthcare application needs a large amount of labeled data of persons. However, in real applications, data are often separate and few people or organizations are willing to disclose their private data. In addition, an increasing number of regulations, such as [9], [10], hold back the leakages of data. These make different clients cannot exchange data directly, and the scattered data forms separate data islands, which makes it impossible to learn a traditional model with aggregated data.

2.2 Federated Learning

Federated learning is a usual way to combine each client's information while protecting data privacy and security [11]. It was first proposed by Google [13], where they proposed FedAvg to train machine learning models via aggregating distributed mobile phones' information without exchanging data. The key idea is to replace direct data exchanges with model parameter-related exchanges. FedAvg is able to resolve the data islanding problems.

Although federated learning is an emerging field, it has attracted much attention [28], [29]. Federated learning can be divided into horizontal federated learning, vertical federated learning, and federated transfer learning according to the characteristics data. When the client features of the two datasets overlap a lot but the clients overlap little, horizontal federated learning can be applied [13]. In the horizontal federated learning, datasets are split horizontally and the clients share the same features finally. For example, Smith et al. [30] proposed a novel systems-aware optimization method, MOCHA, to solve security problems in multitasking. When the client features of the two datasets overlap little but the clients overlap a lot, we can utilize vertical federated learning, where different clients have different columns of the features [31]. For example, Cheng et al. [32] proposed a novel lossless privacy-preserving tree-boosting system known as SecureBoost to jointly conduct over multiple parties with partially common client samples but different feature sets. When the clients and client features of the two datasets both rarely overlap, federated transfer learning is often utilized [33], [34]. For example, Yoon et al. [35] proposed a novel federated continual learning framework, Federated Weighted Inter-client Transfer (FedWeIT), which decomposed the network weights into global federated parameters and sparse task-specific parameters. In [35], each client received selective knowledge from other clients by taking a weighted combination of their task-specific parameters. In addition, many methods, such as differential privacy, are proposed to protect data further [36], [37]. In this paper, we mainly focus on horizontal federated learning when the training data are not independent and identically distributed (Non-IID) on the clients.

Although FedAvg works well in many situations, it may still suffer from the non-iid data and fail to build personalized models for each client [30], [38], [39]. A survey about federated Learning on non-iid Data can be found here [40]. FedProx [17] tackled data non-iid by allowing partial information aggregation and adding a proximal term to FedAvg. [41] aggregated the models of the clients with weights computed via L_1 distance among client models' parameters. These works focus on a common model shared by all clients while some other works try to obtain a unique model for each client. [42] exchanged information of base layers and preserved personalization layer to combat the ill-effects of non-iid. [43] utilized Moreau envelopes as clients' regularized loss function and decoupled personalized model optimization from the global model learning in a bi-level problem stylized for personalized FL. [44] evaluated three techniques for local adaptation of federated models: fine-tuning, multi-task learning, and knowledge distillation. [45] also proposed and analyzed three approaches: user clustering, data interpolation, and model interpolation. [46] tried to jointly learn compact local representations on each device and a global model across all devices with a theoretic analysis. [47] proposed APFL where each client would train

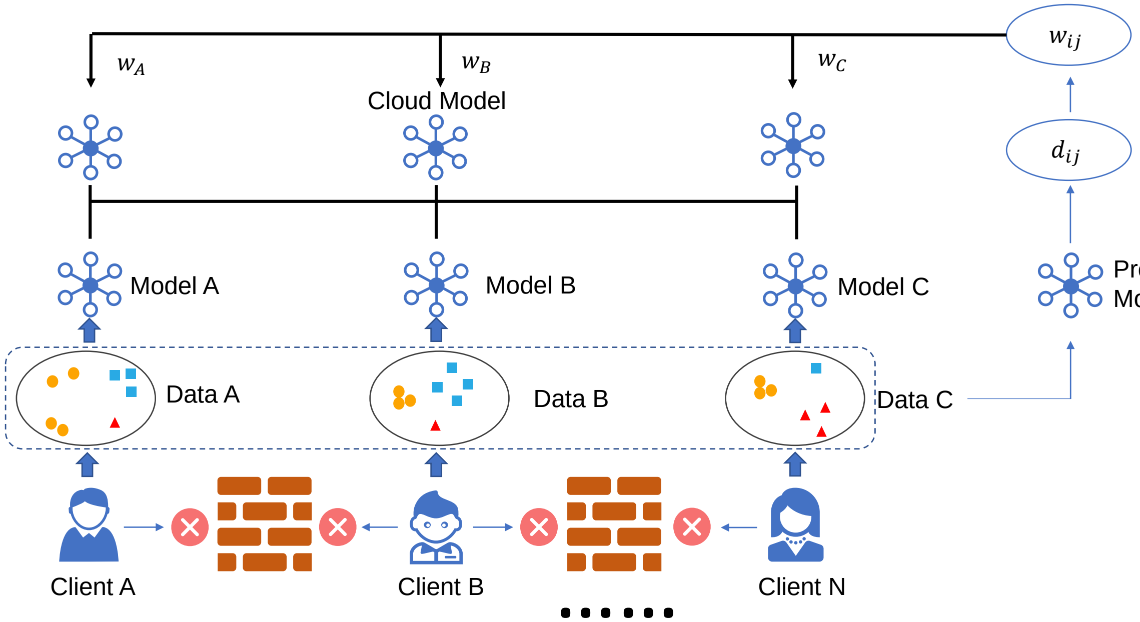


Fig. 2. The structure of the FedAP method.

their local models while contributing to the global model. Another work [48], Clustered Federated Learning (CFL), grouped the client population into clusters with jointly trainable data distributions. Two works most relevant to our method are FedHealth [18] and FedBN [19]. FedHealth needs to share some datasets with all clients while FedBN used local batch normalization to alleviate the feature shift before averaging models. Although there are already some works to cope with data non-iid, few works pay attention to feature shift non-iid and other shifts at the same time and obtaining an individual model for each client in healthcare.

2.3 Batch Normalization

Batch Normalization (BN) [49] is an important component of deep learning. Batch Normalization improves the performance of the model and has a natural advantage in dealing with domain shifts. Li et al. [50] proposed an adaptive BN for domain adaptation where they learned domain-specific BN layers. Nowadays, researchers have explored many effects of BN, especially in transfer learning [51]. FedBN [19] is one of few applications of BN in the field of FL field. However, FedBN does still not make full use of BN properties, and it does not consider the similarities among the clients.

3 Method

3.1 Problem Formulation

In federated learning, there are N different clients (organizations or users), denoted as $\{C_1, C_2, \cdots, C_N\}$ and each client has its own dataset, i.e. $\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N\}$. Each dataset $\mathcal{D}_i = \{(\mathbf{x}_{i,j}, y_{i,j})\}_{j=1}^{n_i}$ contains two parts, i.e. a train dataset $\mathcal{D}_i^{tr} = \{(\mathbf{x}_{i,j}^{tr}, y_{i,j}^{tr})\}_{j=1}^{n_i^{tr}}$ and a test dataset $\mathcal{D}_i^{te} = \{(\mathbf{x}_{i,j}^{te}, y_{i,j}^{te})\}_{j=1}^{n_i^{te}}$. Obviously, $n_i = n_i^{tr} + n_i^{te}$ and $\mathcal{D}_i = \mathcal{D}_i^{tr} \cup \mathcal{D}_i^{te}$. All of the datasets have different distributions, i.e. $P(\mathcal{D}_i) \neq P(\mathcal{D}_j)$. Each client has its own model denoted as $\{f_i\}_{i=1}^N$. Our goal is to aggregate information of all clients to learn a good model f_i for each client on its local dataset $\mathcal{D}_i$ without private data leakage:

$$\min_{\{f_k\}_{k=1}^N} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i^{te}} \sum_{j=1}^{n_i^{te}} \ell(f_i(\mathbf{x}_{i,j}^{te}), y_{i,j}^{te}), \tag{1}$$

where ℓ is a loss function.

3.2 Motivation

There are mainly two challenges for personalized healthcare: data islanding and personalization. Following FedAvg [13] and some other traditional federated learning methods [52], [53], it is easy to cope with the first challenge. Personalization is a must in many applications, especially in healthcare. It is better to train a unique model in each client for personalization. However, one client often lacks enough data to train a model with high accuracy in federated learning. In addition, clients do not have access to the data of other clients. Overall, it is a challenge that how to achieve personalization to obtain high accuracy in federated learning. As mentioned in [50], batch normalization (BN) layers contain sufficient statistics (including mean and standard deviation) of features (outputs of layers). Therefore, BN has been utilized to represent distributions of training data indirectly in many works [50], [54]. We mainly use BN to represent the distributions of clients. Therefore, on the one hand, we utilize local BN to preserve clients' feature distributions. On the other hand, we also use BN-related statistics to calculate the similarity between clients for better personalization with weighted aggregation[1].

[1] Please note that we only share the statistics of the batch normalization layers, and no existing work shows that any methods can restore the specific sample with the statistics of certain layers [55], [56].

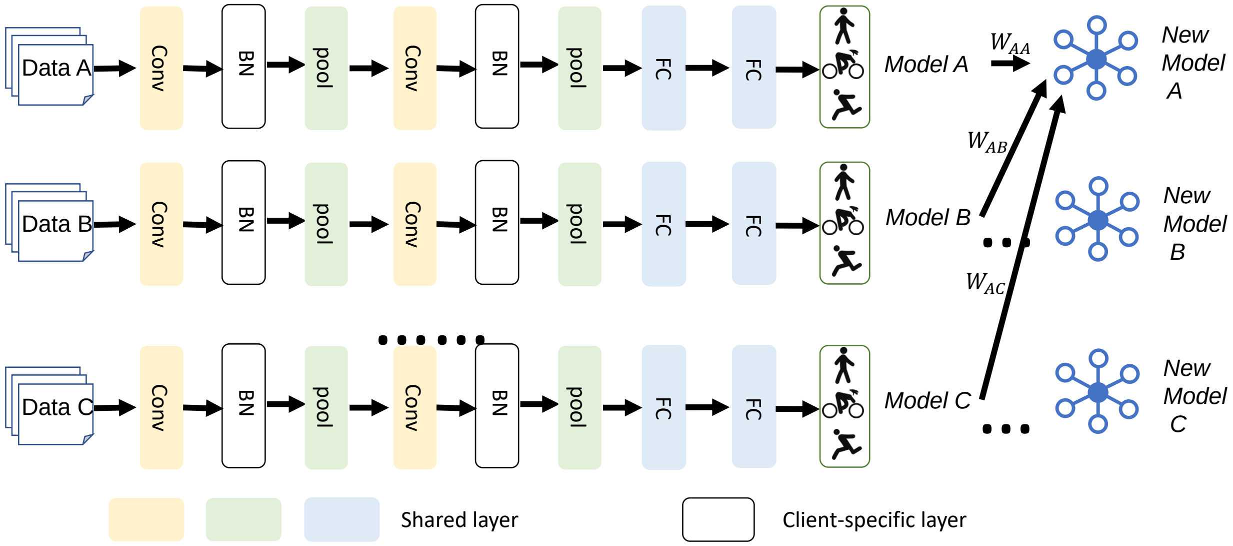


Fig. 3. The concrete process of the FedAP.

3.3 Our Approach: FedAP

In this paper, we propose FedAP (Adaptive Federated Learning) to achieve accurate personal healthcare via adaptive batch normalization without compromising data privacy and security. Fig. 2 gives an overview of its structure. Without loss of generality, we assume there are three clients, which can be extended to more general cases. The structure mainly contains five steps:

哪来的啊

1) The server distributes the pre-trained model to each client.
2) Each client computes statistics of the outputs of specific layers according to local data.
3) The server obtains the client similarities denoted by weight matrix $\mathbf{W}$ to guide aggregation.
4) Each client updates its own model with the local train data and pushes its model to the server.
5) The server aggregates models and obtains N models delivered to N clients respectively.

For stability and simplicity, we only calculate $\mathbf{W}$ once and we show that computing once is enough to achieve acceptable performance in experiments. Note that all processes do not involve the direct transmission of data, so FedAP avoids the leakage of private data and ensures security. The keys of FedAP are obtaining $\mathbf{W}$ and aggregating the models. We will introduce how to compute $\mathbf{W}$ after describing the process of model aggregation.

We denote the parameters of each model f_i as $\boldsymbol{\theta}_i = \boldsymbol{\phi}_i \cup \boldsymbol{\psi}_i$, where $\boldsymbol{\phi}_i$ corresponds to the parameters of BN layers specific to each client and $\boldsymbol{\psi}_i$ is the parameters of the other layers (colored blocks in Fig. 3). $\mathbf{W}$ is an $N \times N$ matrix, which describes the similarities among the clients. $w_{ij} \in [0, 1]$ represents the similarity between client i and client j: the larger w_{ij} is, the more similar the two clients are.

Fig. 3 demonstrates the process of model aggregation. As shown in Fig. 3, $\boldsymbol{\phi}_i$ is particular while $\boldsymbol{\psi}_i$ is computed according to $\mathbf{w}_i$, where $\mathbf{w}_i$ means the $i-$th row of $\mathbf{W}$, and $\boldsymbol{\psi}$, where $\boldsymbol{\psi} = \{\boldsymbol{\psi}_i\}_{i=1}^N$. $\boldsymbol{\phi}_i$ is BN parameters that are not shared across clients while $\boldsymbol{\psi}_i$ is other parameters that are shared.

Algorithm 1 FedAP

```
Input: A pre-trained model $f$, data of $N$ clients $\{\mathcal{D}_i\}_{i=1}^N$, $\lambda$
Output: Client models $\{f_i\}_{i=1}^N$
1: Distribute $f$ to each client
2: Each client computes its statistics $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$, where $\boldsymbol{\mu}_i$ represents the mean values while $\boldsymbol{\sigma}_i$ represents the covariance matrices. Push $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ to the server
3: Compute $\mathbf{W}$ according to the statistics
4: Update clients' model with local data. Push updated parameter $\{\boldsymbol{\theta}_i^{t*}\}_{i=1}^N$ to the server
5: Update $\{\boldsymbol{\theta}_i^{t+1}\}_{i=1}^N$ according to Eq. (2) and distribute them to the corresponding clients
6: Repeat steps $4 \sim 5$ until convergence or maximum round reached
```

Let $\boldsymbol{\theta}_i^t = \boldsymbol{\phi}_i^t \cup \boldsymbol{\psi}_i^t$ represent the parameters of the model from client i in the round t. After updating $\boldsymbol{\theta}_i^t$ with the local data from the $i-$th client, we obtain updated parameters $\boldsymbol{\theta}_i^{t*} = \boldsymbol{\phi}_i^{t*} \cup \boldsymbol{\psi}_i^{t*}$. We use the $*$ notation to denote updated parameters. Then, for aggregation on the server, we have the following updating strategy:

$$\begin{cases} \boldsymbol{\phi}_i^{t+1} = & \boldsymbol{\phi}_i^{t*} \\ \boldsymbol{\psi}_i^{t+1} = & \sum_{j=1}^N w_{ij} \boldsymbol{\psi}_j^{t*}. \end{cases} \tag{2}$$

The overall process of FedAP is described in Algorithm 1. In the next sections, we will introduce how to compute the weight matrix $\mathbf{W}$.

3.4 Evaluate Weights

In this section, we will evaluate the weights with a pre-trained model f and propose two alternatives to compute the weights. We mainly rely on the feature output statistics of clients' data in the pre-trained network to compute weights.

We denote with $l \in \{1, 2, \cdots, L\}$ in superscript notations the different batch normalization layers in the model. And $\mathbf{z}^{i,l}$ represents the input of $l-$th batch normalization layer in the $i-$th client. The input of the classification layer in the $i-$th client is denoted as $\mathbf{z}^i$ which represents the domain features. We assume $\mathbf{z}^{i,l}$ is a matrix, $\mathbf{z}^{i,l}_{c_{i,l}\times s_{i,l}}$ where $c_{i,l}$ corresponds to the channel number while $s_{i,l}$ is the product of the other dimensions. Similarly, $\mathbf{z}^i = \mathbf{z}^i_{c_i \times s_i}$. We feed $\mathcal{D}_i$ into f, and we can obtain $\mathbf{z}^{i,l}_{c_{i,l}\times s_{i,l}}$. Obviously, $s_{i,l} = e \times n_i$ where e is an integer. Now, we try to compute statistics on the channels, and we treat $\mathbf{z}^{i,l}$ as a Gaussian distribution. For the $l-$th layer of the $i-$th client, it is easy to obtain its distribution, $\mathcal{N}(\boldsymbol{\mu}^{i,l}, \boldsymbol{\sigma}^{i,l})$. We only compute statistics of inputs of BN layers. And the BN statistics of the $i-$th client is formulated as:

$$(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) = [(\boldsymbol{\mu}^{i,1}, \boldsymbol{\sigma}^{i,1}), (\boldsymbol{\mu}^{i,2}, \boldsymbol{\sigma}^{i,2}), \cdots, (\boldsymbol{\mu}^{i,L}, \boldsymbol{\sigma}^{i,L})]. \quad (3)$$

Now we can calculate the similarity between two clients. It is popular to adopt the Wasserstein distance to calculate the distance between two Gaussian distributions:

$$\begin{aligned} &W_2^2(\mathcal{N}(\boldsymbol{\mu}^{i,l}, \boldsymbol{\sigma}^{i,l}), \mathcal{N}(\boldsymbol{\mu}^{j,l}, \boldsymbol{\sigma}^{j,l})) \\ =&||\boldsymbol{\mu}^{i,l} - \boldsymbol{\mu}^{j,l}||^2 + \\ &tr(\boldsymbol{\sigma}^{i,l} + \boldsymbol{\sigma}^{j,l} - 2((\boldsymbol{\sigma}^{i,l})^{1/2}\boldsymbol{\sigma}^{j,l}(\boldsymbol{\sigma}^{i,l})^{1/2})^{1/2}), \end{aligned} \quad (4)$$

where tr is the trace of the matrix. Obviously, it is costly and difficult to perform efficient calculations. Similar to BN, we perform approximations and consider that each channel is independent of the others. Therefore, $\boldsymbol{\sigma}^{i,l}$ is a diagonal matrix, i.e. $\boldsymbol{\sigma}^{i,l} = Diag(\mathbf{r}^{i,l})$. Therefore, we compute the approximation of Wasserstein distance as:

$$\begin{aligned} &W_2^2(\mathcal{N}(\boldsymbol{\mu}^{i,l}, \boldsymbol{\sigma}^{i,l}), \mathcal{N}(\boldsymbol{\mu}^{j,l}, \boldsymbol{\sigma}^{j,l})) \\ =&||\boldsymbol{\mu}^{i,l} - \boldsymbol{\mu}^{j,l}||^2 + ||\sqrt{\mathbf{r}^{i,l}} - \sqrt{\mathbf{r}^{j,l}}||_2^2. \end{aligned} \quad (5)$$

Thus, the distance between two clients i, j is computed as:

$$\begin{aligned} d_{i,j} &= \sum_{l=1}^{L} W_2(\mathcal{N}(\boldsymbol{\mu}^{i,l}, \boldsymbol{\sigma}^{i,l}), \mathcal{N}(\boldsymbol{\mu}^{j,l}, \boldsymbol{\sigma}^{j,l})) \\ &= \sum_{l=1}^{L} (||\boldsymbol{\mu}^{i,l} - \boldsymbol{\mu}^{j,l}||^2 + ||\sqrt{\mathbf{r}^{i,l}} - \sqrt{\mathbf{r}^{j,l}}||_2^2)^{1/2}. \end{aligned} \quad (6)$$

Large $d_{i,j}$ means the distribution distance between the $i-$th client and the jth client is large. Therefore, the larger $d_{i,j}$ is, the less similar the two clients are, which means the smaller $w_{i,j}$ is. And we set $\tilde{w}_{i,j}$ as the inverse of $d_{i,j}$, i.e. $\tilde{w}_{i,j} = 1/d_{i,j}, j \neq i$. Normalize $\tilde{w}_i$ and we have

$$\hat{w}_{i,j} = \frac{\tilde{w}_{i,j}}{\sum_{j=1, j\neq i}^{N} \tilde{w}_{i,j}}, \text{ where } j \neq i \quad (7)$$

For stability in training, we take $\boldsymbol{\psi}^{t*}$ into account for $\boldsymbol{\psi}^{t+1}$. We update $\boldsymbol{\psi}^{t+1}$ in a moving average style, and we set $w_{i,i} = \lambda$. Therefore,

$$w_{i,j} = \begin{cases} \lambda, & i = j, \\ (1-\lambda) \times \hat{w}_{i,j}, & i \neq j. \end{cases} \quad (8)$$

We denote this weighting method as the original *FedAP*. Similarly, we can obtain the corresponding $\mathbf{W}$ using only the last layer $\mathbf{z}^i$ and we denote this variant as *d-FedAP*.

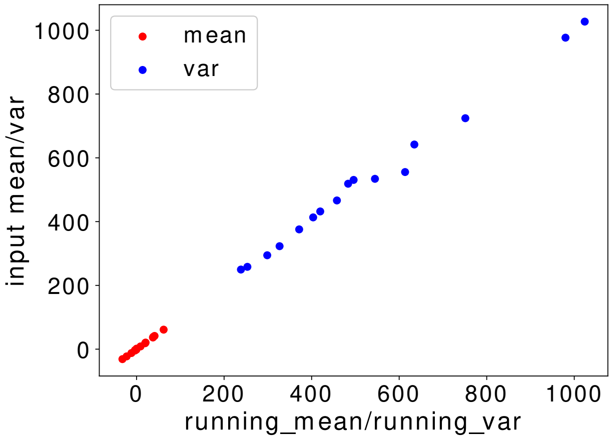


Fig. 4. Running mean, running var of a BN layer and the inputs statistics of the corresponding layer in a client model.

3.5 Discussion

In some extreme cases, there may not exist a pre-trained model. In this situation, we can evaluate weights with models trained from several rounds of FedBN [19].

As we can see from Fig. 4, the running mean of the BN layer has a positive correlation with the statistical mean of the corresponding layer's inputs. And the variance has a similar relationship. From this, we can use running means and running variances of the BN layers instead of the statistics respectively. Therefore, we can perform several rounds of FedBN [19] which preserves local batch normalization, and utilize parameters of BN layers to replace the statistics when there does not exist a pre-trained model. We denote this variant as *f-FedAP*.

4 Experiments

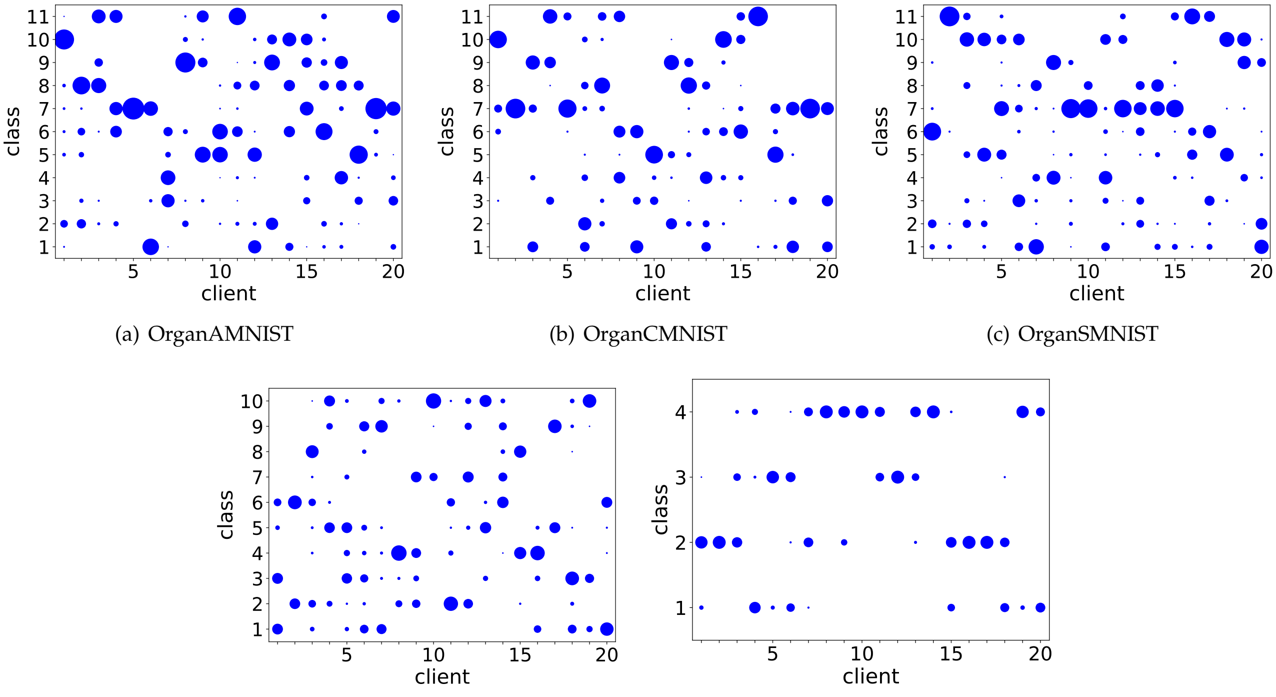
We evaluate the performance of FedAP on five healthcare datasets in time series and image modalities[2]. The statistical information of each dataset is shown in TABLE 1.

4.1 Datasets

PAMAP2. We adopt a public human activity recognition dataset called PAMAP2 [57]. The PAMAP2 dataset contains data of 18 different physical activities, performed by 9 subjects wearing 3 inertial measurement units and a heart rate monitor. We use data of 3 inertial measurement units which are collected at a constant rate of 100Hz to form data containing 27 channels. We exploit the sliding window technique and filter out 10 classes of data[3]. In order to construct the problem situation in FedAP, we use the Dirichlet distribution as in [58] to create disjoint non-iid splits. client training data. Fig. 5(d) visualizes how samples are distributed among 20 clients. In each client, half of the

[2] Code is released at https://github.com/jindongwang/tlbook-code/tree/main/chap19_fl and https://github.com/microsoft/PersonalizedFL.

[3] We split PAMAP2 in this style mainly for two reasons. On the one hand, the data numbers of the subjects are different which may introduce some other problems, e.g. some clients cannot be adequately evaluated. On the other hand, this splitting routing is widely adopted in much work [58], [59]. We select 10 classes with the most samples.



(a) OrganAMNIST

(b) OrganCMNIST

(c) OrganSMNIST

(d) PAMAP2

(e) COVID-19

Fig. 5. The number of samples per class allocated to each client (indicated by dot size).

data are used to train and the remaining data are for testing as in [19].

COVID-19. We also adopt a public COVID-19 posterior-anterior chest radiography images dataset [60]. This is a combined curated dataset of COVID-19 Chest X-ray images obtained by collating 15 public datasets and it contains 9,208 instances of four classes (1,281 COVID-19 X-Rays, 3,270 Normal X-Rays, 1,656 viral-pneumonia X-Rays, and 3,001 bacterial-pneumonia X-Rays) in total. In order to construct the problem situation in FedAP, we split the dataset similar to PAMAP2. Fig. 5(e) visualizes how samples are distributed among 20 clients for COVID-19. Note that this dataset is more unbalanced in classes which is an ideal testbed to test the performance under label shift (i.e., imbalanced class distribution for different clients). In each client, half of the data are used to train and the remaining data are for testing.

MedMnist. MedMNIST [61], [62] is a large-scale MNIST-like collection of standardized biomedical images, including 12 datasets for 2D and 6 datasets for 3D. All images are 28 × 28 (2D) or 28 × 28 × 28 (3D). We choose 3 datasets which have most classes from 12 2D datasets: OrganAMNIST, OrganCMNIST, OrganSMNIST [63], [64]. These three datasets are all about Abdominal CT images and all contain 11 classes. There are 58,850, 23,660 and 25,221 samples respectively. As operations in PAMAP2, each dataset is split into 20 clients with Dirichlet distributions, and Fig. 5(a)-5(c) visualizes how samples are distributed for OrganAMNIST, OrganCMNIST, and OrganSMNIST respectively. In each client, half of the data are used to train and the remaining data are for testing.

TABLE 1
Statistical information of five datasets.

Dataset	Type	#Class	#Sample
PAMAP2	Sensor-based time series	18	3,850,505
COVID-19	Image	4	9,208
OrganAMNIST	Image	11	58,850
OrganCMNIST	Image	11	23,660
OrganSMNIST	Image	11	25,221

4.2 Implementations Details and Comparison Methods

For PAMAP2, we adopt a CNN for training and predicting. The network is composed of two convolutional layers, two pooling layers, two batch normalization layers, and two fully connected layers. For three MedMNIST datasets, we all adopt LeNet5 [65]. For COVID-19, we adopt Alexnet [66]. We use a three-layer fully connected neural network as the classifier with two BN layers after the first two fully connected layers following [19]. For model training, we use the cross-entropy loss and SGD optimizer with a learning rate of 10^{-2}. If not specified, our default setting for local update epochs is $E = 1$ where E means training epochs in one round. And we set $\lambda = 0.5$ for our method, since we can see that λ has few influences on accuracy and it only affects convergence speeds in the appendix. In addition, we randomly select 20% of the data to train a model of the same architecture as the pre-trained model. We run three trials to record the average results.

We compare three extensions of our method with five methods including common FL methods and some FL methods designed for non-iid data:

- Base: Each client uses local data to train its local models

TABLE 2
Activity recognition results on PAMAP2. Bold and underline mean the best and second-best results, respectively.

Client	Base	FedAvg	FedBN	FedProx	FedPer	FedAP
1	**92.86**	60.27	60.72	60.5	48.31	77.2
2	17.68	62.36	62.59	62.36	**97.51**	77.55
3	**100**	50.56	50.34	50.34	61.4	77.43
4	**83.52**	73.98	73.53	73.98	47.29	79.64
5	18.78	74.27	74.72	73.81	58.47	**81.94**
6	77.66	62.9	62.44	61.76	23.98	**79.86**
7	**95.05**	64.03	62.9	63.57	49.55	86.2
8	17.58	87.78	88.24	87.78	91.86	**95.02**
9	**92.39**	74.49	74.27	74.27	51.24	85.33
10	**93.37**	64.71	64.48	64.71	77.6	69.23
11	29.12	65.24	65.69	66.37	89.16	**91.42**
12	**84.78**	63.35	62.9	63.12	57.92	79.41
13	**98.9**	68.33	68.33	68.33	42.53	74.43
14	24.18	64.79	65.24	65.69	49.44	**69.75**
15	**98.91**	63.12	62.44	62.44	58.6	81.67
16	**98.9**	85.26	85.94	85.49	86.62	94.1
17	41.44	66.21	65.99	66.21	77.32	**82.77**
18	**93.62**	59.64	59.64	59.41	52.38	75.74
19	**85.71**	67.87	68.1	67.87	73.08	77.15
20	37.02	72.46	72.69	72.46	**97.52**	86.91
avg	69.07	67.58	67.56	67.52	64.59	**81.14**

TABLE 3
Accuracy on OrganAMNIST. Bold and underline mean the best and second-best results, respectively.

Client	Base	FedAvg	FedBN	FedProx	FedPer	FedAP
1	48.35	80.03	**96.06**	80.37	83.22	81.86
2	55.25	92.46	93.14	92.26	78.68	**94.5**
3	34.04	86.15	96.27	86.22	71.08	**97.08**
4	61.54	77.65	87.91	77.58	41.71	**88.52**
5	41.44	92.32	**100**	92.66	**100**	99.93
6	52.13	84.38	**97.55**	84.92	80.57	95.52
7	42.31	83.42	50.07	82.95	64.06	**87.84**
8	48.9	92.81	**97.15**	92.94	87.86	96.95
9	38.04	74.66	84.1	74.39	62.16	**85.46**
10	38.12	72.27	82.98	72.34	53.22	**87.53**
11	59.89	74.88	90.36	74.88	66.33	**91.79**
12	59.78	78.41	**90.56**	78.28	72.84	89.75
13	44.75	91.17	**97.69**	91.04	78.14	97.42
14	52.2	83.76	**92.26**	83.9	61.28	90.29
15	53.26	89.61	87.84	90.42	58.29	**94.29**
16	70.88	83.31	**93.62**	83.18	73.54	91.59
17	36.46	92.93	62.77	92.93	45.92	**94.97**
18	46.81	77.99	**96.94**	77.92	78.12	96.67
19	31.32	92.05	**96.94**	92.26	93.95	96.47
20	45.3	81.02	91.32	80.75	49.36	**93.97**
avg	48.04	84.06	89.28	84.11	70.02	**92.62**

TABLE 4
Accuracy on OrganCMNIST. Bold and underline mean the best and second-best results, respectively.

Client	Base	FedAvg	FedBN	FedProx	FedPer	FedAP
1	32.61	79.22	90.54	79.73	77.87	**94.59**
2	52.17	95.61	**100**	95.95	**100**	**100**
3	47.1	85.83	88.7	85.83	72.85	**93.93**
4	37.23	84.34	**96.97**	84.01	74.41	96.3
5	48.91	92.41	96.46	92.24	84.49	**97.13**
6	51.45	65.6	75.89	65.6	53.96	**85.5**
7	64.49	86.22	86.72	86.55	76.64	**89.08**
8	45.65	50.93	61.38	50.59	38.45	**92.07**
9	26.09	81.28	**91.23**	80.61	45.03	86.68
10	57.25	54.56	**93.41**	54.9	80.07	91.89
11	54.89	79.73	**94.76**	79.73	83.78	91.39
12	50.72	90.56	**95.78**	90.39	67.28	89.38
13	66.67	49.33	54.71	49.16	52.36	**89.73**
14	37.16	74.32	**88.34**	73.99	70.61	87.33
15	58.57	75.93	86.36	75.59	51.52	**89.9**
16	52.9	81.25	98.82	80.91	**98.99**	98.82
17	50	67.45	**83.81**	67.45	70.66	81.11
18	52.9	88.18	90.71	88.18	65.37	**91.89**
19	26.28	94.26	**100**	94.26	**100**	99.83
20	59.78	90.25	88.91	90.25	58.49	**93.78**
avg	48.64	78.36	88.18	78.3	71.14	**92.02**

TABLE 5
Accuracy on OrganSMNIST. Bold and underline mean the best and second-best results, respectively.

Client	Base	FedAvg	FedBN	FedProx	FedPer	FedAP
1	25.27	47.39	**91.76**	39.94	85.74	90.65
2	32.04	55.7	**96.04**	62.82	94.78	93.51
3	30.85	63.13	71.84	66.3	52.69	**73.89**
4	41.21	59.65	**78.8**	61.71	64.56	75.32
5	37.02	68.45	80.28	73.03	68.61	**83.12**
6	40.43	48.02	77.02	52.3	58	**80.67**
7	38.46	78.2	83.57	71.25	78.2	**85.47**
8	40.11	57.44	53.48	41.93	**96.52**	96.2
9	42.39	83.73	**94.94**	87.99	92.42	**94.94**
10	24.86	88.45	99.21	87.97	97.47	**99.37**
11	38.46	60.79	45.04	37.17	54.49	**76.06**
12	41.3	79.15	81.67	83.57	78.04	**88.47**
13	43.65	66.98	79.94	67.93	58.29	**80.57**
14	44.51	84.99	**95.73**	86.57	84.52	92.58
15	44.57	80.7	**88.77**	84.49	73.1	87.34
16	48.9	52.06	**73.58**	53.64	71.36	**73.58**
17	36.46	36.55	**78.48**	39.56	61.55	76.42
18	45.74	58.7	**79.43**	60.44	69.3	**79.59**
19	30.77	53.48	86.23	61.87	64.72	**91.46**
20	35.91	72.51	72.99	77.57	**86.57**	68.4
avg	38.15	64.8	80.44	64.9	74.55	**84.38**

without federated learning.

- FedAvg [13]: The server aggregates all client models without any particular operations for non-iid data.
- FedProx [17]: Allow partial information aggregation and add a proximal term to FedAvg.
- FedPer [42]: Each client preserves some local layers.
- FedBN [19]: Each client preserves the local batch normalization.

4.3 Classification Accuracy

The classification results for each client on PAMAP2 are shown in TABLE 2. From these results, we have the following observations: 1) Our method achieves the best results on average. It is obvious that our method significantly outperforms other methods with a remarkable improve-

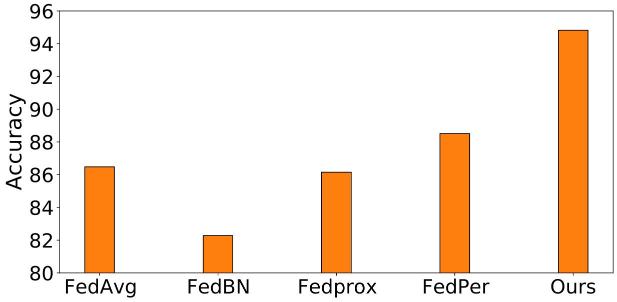
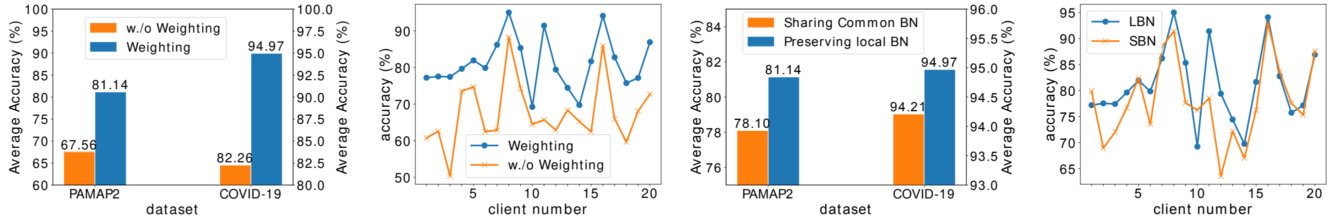
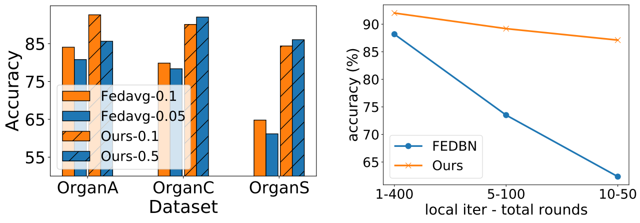


Fig. 6. Average accuracy of 20 clients on COVID-19.



(a) The weighting technique (b) Different clients (c) The local BN sharing (d) Client Acc on PAMAP2

Fig. 7. The effects of weighting and preserving local batch normalization. Each point has equal status in Fig. 7(b) and Fig. 7(d). We use a line chart just for better visualization effects but not trends.



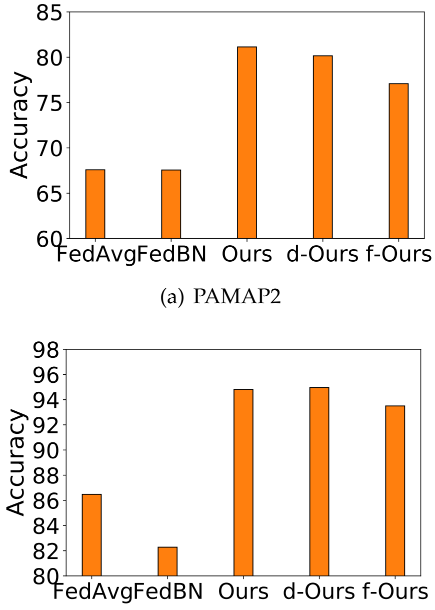
(a) Dataset Split (b) Iteration-Round

Fig. 8. Influence of Dataset split and Iteration-Round.

ment (over $\mathbf{10}\%$ on average). 2) In some clients, the base method achieves the best test accuracy. As it can be seen from Fig. 5(d), the distributions on the clients are very inconsistent, which inevitably leads to the various difficulty levels in different clients. And some distributions in the corresponding clients are so easy that only utilizing the local data can achieve the ideal effects. 3) FedBN does not achieve the desired results. This could be caused by that FedBN is designed for the feature shifts while our experiments are mainly set in the label shifts.

The classification results for each client on three MedMNIST datasets are shown in TABLE 3, 4, 5 respevtively. From these results, we have the following observations: 1) Our method significantly outperforms other methods with a remarkable improvement (over $\mathbf{3.5}\%$ on average). 2) For all these three benchmarks, Base achieves the worst average accuracy, which demonstrates Base without communicating with each other does not have enough information for these relatively difficult tasks. 3) FedBN achieves the second best results on all three benchmarks. This could be because that there exist feature shifts among clients.

The classification results for each client on COVID-19 are shown in Fig. 6. From these results, we have the following observations: 1) Our method achieves the best average accuracy which outperforms the second-best method FedPer by $\mathbf{6.3}\%$ on average accuracy. 2) FedBN gets the worst results. This demonstrates that FedBN is not good at dealing with label shifts where label distributions of each client are different, which is a challenging situation. FedBN does not consider the similarities among different clients. From Fig. 5(e), we can see that label shifts are serious in COVID-19 since it only has four classes.



(a) PAMAP2

(b) COVID

Fig. 9. Evaluating two variants of FedAP.

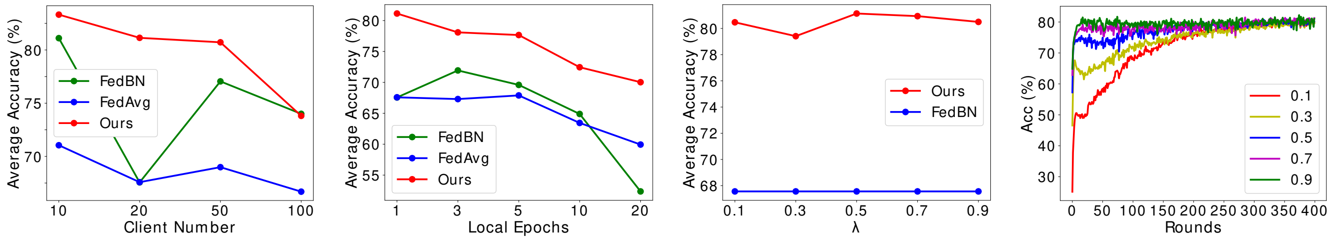
4.4 Analysis and discussion

We consider the influence of data splits and local iterations in this section. As shown in Fig. 8(a), we evaluate Fedavg and FedAP on three MedMNIST benchmarks with two different splits: $\alpha = 0.1$ and $\alpha = 0.05$ respectively. Smaller α means distributions among clients are more different from each other. Fig. 8(a) demonstrates that the performance of Fedavg which does not consider data non-iid will drop when encountering clients with greater different distributions while our method is not affected much by the degree of data non-iid, which means our method may be more robust. Fig. 8(b) shows the influence of local iterations and total rounds on FedBN and our method. It is obvious that FedBN drops seriously with more local iterations and fewer communication rounds while our method declines slowly, which means when limiting communication costs, our method may be more effective.

4.5 Ablation Study

Effects of Weighting.

To demonstrate the effect of weighting which considers the similarities among the different clients, we compare



(a) Client Number (b) Local Epochs (c) λ (d) λ

Fig. 10. Parameter sensitivity analysis.

the average accuracy on PAMAP2 and COVID-19 between the experiments with it and without it. Without weighting, our method degenerates to FedBN. From Fig. 7(a), we can see that our method performs much better than FedBN which does not include the weighting part. Moreover, from Fig. 7(b), we can see our method performs better than FedBN on all clients. These results demonstrate that our method with weighting can cope with the label shifts while FedBN cannot deal with this situation, which means our method is more applicable and effective.

Effects of Preserving Local Batch Normalization.

We illustrate the importance of preserving local batch normalization. Fig. 7(c) shows the average accuracy between the experiments with preserving local batch normalization and the experiments with sharing common batch normalization while Fig. 7(d) shows the results on each client. LBN means preserving local batch normalization while SBN means sharing common batch normalization. Obviously, the improvements are not particularly significant compared with weighting. This may be caused by there mainly exist the label shifts in our experiments while preserving local batch normalization is for the feature shifts. However, our method still has a slight improvement, indicating its superiority.

Different Implementations of Our methods.

In Method section, we propose three implementations of our method: FedAP, d-FedAP, and f-FedAP. The main differences among them are how to calculate $\mathbf{W}$. In Fig. 9(a) and Fig. 9(b), we can see that all three implementations achieve better average accuracy on both PAMAP2 and COVID compared with FedAvg and FedBN. In addition, f-FedAP performs slightly worse than the other two variants, which may be because it only utilizes weighting during half rounds for fairness and the other half are for obtaining $\mathbf{W}$.

4.6 Convergence and Parameter Sensitivity

We study the convergence of our method. From Fig. 11, we can see our method almost convergences in the 10th round. And in the actual experiments, 20 rounds are enough for our method while FedBN needs over 400 rounds.

Then, we evaluate the parameter sensitivity of FedAP. Our method is affected by three parameters: local epochs, client number, and λ. We change one parameter and fix the other parameters.

From Fig. 10(a), we can see that our method still achieves acceptable results. When the client numbers increase, our method goes down which may be due to that few data in local clients make the weight estimation inaccurate. And we may take f-FedAP instead. In Fig. 10(b), we can see our method is the best and it is descending with local epochs increasing, which may be caused that we keep the total number of the epochs unchanged and the communication among the clients are insufficient. Fig. 10(c)-10(d) demonstrates λ slightly affects the average accuracy of our method while it can change the convergence rate. The results reveal that FedAP is more effective and robust than other methods under different parameters in most cases.

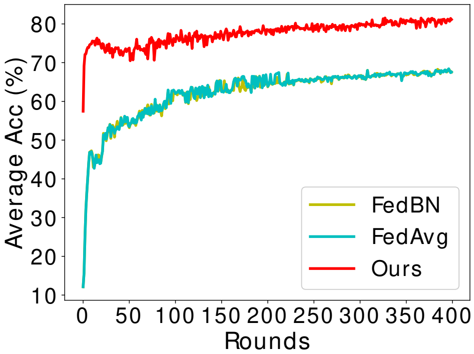


Fig. 11. Convergence analysis of different methods.

5 Conclusions and Future Work

In this article, we proposed FedAP, a weighted personalized federated transfer learning algorithm via batch normalization for healthcare. FedAP aggregates the data from different organizations without compromising privacy and security and achieves relatively personalized model learning through combing considering similarities and preserving local batch normalization. Experiments have evaluated the effectiveness of FedAP. In the future, we plan to apply FedAP to more personalized and flexible healthcare. And we will consider better ways to calculate and update similarities among clients.

Acknowledgement

This work is supported by the National Key Research and Development Plan of China No.2021YFC2501202, Natural Science Foundation of China (No. 61972383, No. 61902377, No. 61902379), Beijing Municipal Science & Technology Commission No.Z211100002121171.

References

[1] S. Sharma, V. Elvira, E. Chouzenoux, and A. Majumdar, "Recurrent dictionary learning for state-space models with an application in stock forecasting," *Neurocomputing*, vol. 450, pp. 1–13, 2021.

[2] B. Brattoli, J. Tighe, F. Zhdanov, P. Perona, and K. Chalupka, "Rethinking zero-shot video classification: End-to-end training for realistic applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4613–4623.

[3] W. Lu, J. Wang, and Y. Chen, "Local and global alignments for generalizable sensor-based human activity recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3833–3837.

[4] W. Lu, Y. Chen, J. Wang, and X. Qin, "Cross-domain activity recognition via substructural optimal transport," *Neurocomputing*, vol. 454, pp. 65–75, 2021.

[5] Z. Fei, E. Yang, D. D.-U. Li, S. Butler, W. Ijomah, X. Li, and H. Zhou, "Deep convolution network based emotion analysis towards mental health care," *Neurocomputing*, vol. 388, pp. 212–227, 2020.

[6] E. Choi, C. Xiao, J. Sun, and W. F. Stewart, "Mime: Multilevel medical embedding of electronic health records for predictive healthcare," vol. 2018, pp. 4547–4557, 2018.

[7] A. B. Ünal, M. Akgün, and N. Pfeifer, "Escaped: Efficient secure and private dot product framework for kernel-based machine learning algorithms with applications in healthcare," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9988–9996.

[8] W. Lu, J. Wang, Y. Chen, S. Pan, C. Hu, and X. Qin, "Semantic-discriminative mixup for generalizable sensor-based cross-domain activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable, and Ubiquitous Technologies*, 2022.

[9] N. Inkster, *China's cyber power*. Routledge, 2018.

[10] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, p. 3152676, 2017.

[11] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[12] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, no. 1, pp. 1–19, 2021.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[14] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman *et al.*, "Opportunities and obstacles for deep learning in biology and medicine," *Journal of The Royal Society Interface*, vol. 15, no. 141, p. 20170387, 2018.

[15] W. Zhu, L. Xie, J. Han, and X. Guo, "The application of deep learning in cancer prognosis prediction," *Cancers*, vol. 12, no. 3, p. 603, 2020.

[16] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations*, 2019.

[17] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. mlsys.org, 2020.

[18] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[19] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," in *International Conference on Learning Representations (ICLR)*, 2021.

[20] Z. Chen, J. Zhang, S. Che, J. Huang, X. Han, and Y. Yuan, "Diagnose like a pathologist: Weakly-supervised pathologist-tree network for slide-level immunohistochemical scoring," in *35th AAAI Conference on Artificial Intelligence (AAAI-21)*. AAAI Press, 2021, pp. 47–54.

[21] I. Avellino, G. Bailly, G. Canlorbe, J. Belgihti, G. Morel, and M.-A. Vitrani, "Impacts of telemanipulation in robotic assisted surgery," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–15.

[22] G. Muhammad, M. S. Hossain, and N. Kumar, "Eeg-based pathology detection for home health monitoring," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 603–610, 2020.

[23] W. Ji, S. Yu, J. Wu, K. Ma, C. Bian, Q. Bi, J. Li, H. Liu, L. Cheng, and Y. Zheng, "Learning calibrated medical image segmentation via multi-rater agreement modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 341–12 351.

[24] F. Sun, W. Zang, R. Gravina, G. Fortino, and Y. Li, "Gait-based identification for elderly users in wearable healthcare systems," *Information fusion*, vol. 53, pp. 134–144, 2020.

[25] Y. Chen, X. Yang, B. Chen, C. Miao, and H. Yu, "Pdassist: Objective and quantified symptom assessment of parkinson's disease via smartphone," in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2017, pp. 939–945.

[26] F. R. Vogenberg, C. I. Barash, and M. Pursel, "Personalized medicine: part 1: evolution and development into theranostics," *Pharmacy and Therapeutics*, vol. 35, no. 10, p. 560, 2010.

[27] I. Chung, S. Kim, J. Lee, K. J. Kim, S. J. Hwang, and E. Yang, "Deep mixed effect model using gaussian processes: A personalized and reliable prediction for healthcare," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3649–3657.

[28] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5476–5497, 2020.

[29] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[30] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 4427–4437.

[31] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *arXiv preprint arXiv:1711.10677*, 2017.

[32] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang, "Secureboost: A lossless federated learning framework," *IEEE Intelligent Systems*, 2021.

[33] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, 2020.

[34] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large cnns at the edge," vol. 33, 2020.

[35] J. Yoon, W. Jeong, G. Lee, E. Yang, and S. J. Hwang, "Federated continual learning with weighted inter-client transfer," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 073–12 086.

[36] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[37] N. Agarwal, A. T. Suresh, F. Yu, S. Kumar, and H. B. McMahan, "cpsgd: communication-efficient and differentially-private distributed sgd," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 7575–7586.

[38] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar, "Adaptive gradient-based meta-learning methods," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 5917–5928.

[39] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, vol. 139. PMLR, 2021, pp. 12 878–12 889.

[40] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: A survey," *arXiv preprint arXiv:2106.06843*, 2021.

[41] Y. Yeganeh, A. Farshad, N. Navab, and S. Albarqouni, "Inverse distance aggregation for federated learning with non-iid data," in *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*. Springer, 2020, pp. 150–159.

[42] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[43] C. T Dinh, N. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[44] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging federated learning by local adaptation," *arXiv preprint arXiv:2002.04758*, 2020.

[45] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arXiv preprint arXiv:2002.10619*, 2020.

[46] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," *arXiv preprint arXiv:2001.01523*, 2020.

[47] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

[48] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 8, pp. 3710–3722, 2020.

[49] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[50] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognition*, vol. 80, pp. 109–117, 2018.

[51] M. Segù, A. Tonioni, and F. Tombari, "Batch normalization embeddings for deep domain generalization," *arXiv preprint arXiv:2011.12672*, 2020.

[52] H. Gao, A. Xu, and H. Huang, "On the convergence of communication-efficient local sgd for federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 7510–7518.

[53] X. Cao, J. Jia, and N. Z. Gong, "Provably secure federated learning against malicious clients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6885–6893.

[54] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7354–7362.

[55] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[56] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 337–16 346.

[57] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*. IEEE, 2012, pp. 108–109.

[58] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7252–7261.

[59] T. Lin, S. P. Karimireddy, S. Stich, and M. Jaggi, "Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6654–6665.

[60] U. Sait, K. G. Lal, S. Prajapati, R. Bhaumik, T. Kumar, S. Sanjana, and K. Bhalla, "Curated dataset for covid-19 posterior-anterior chest radiography images (x-rays)," *Mendeley Data*, vol. 1, 2020.

[61] J. Yang, R. Shi, and B. Ni, "Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis," in *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 2021, pp. 191–195.

[62] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification," *arXiv preprint arXiv:2008.#TODO*, 2021.

[63] P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen, Q. Dou, C.-W. Fu, X. Han, P.-A. Heng, J. Hesser *et al.*, "The liver tumor segmentation benchmark (lits)," *arXiv preprint arXiv:1901.04056*, 2019.

[64] X. Xu, F. Zhou, B. Liu, D. Fu, and X. Bai, "Efficient multiple organ localization in ct image using 3d region proposal network," *IEEE transactions on medical imaging*, vol. 38, no. 8, pp. 1885–1898, 2019.

[65] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[66] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, vol. 25, 2012, pp. 1097–1105.