Medical Federated Model with Mixture of Personalized and Shared Components

Yawei Zhao, Qinghe Liu†, Pan Liu, Xinwang Liu‡, Kunlun He‡

***Abstract*—Although data-driven methods usually have noticeable performance on disease diagnosis and treatment, they are suspected of leakage of privacy due to collecting data for model training. Recently, federated learning provides a secure and trustable alternative to collaboratively train model without any exchange of medical data among multiple institutes. Therefore, it has draw much attention due to its natural merit on privacy protection. However, when heterogenous medical data exists between different hospitals, federated learning usually has to face with degradation of performance. In the paper, we propose a new personalized framework of federated learning to handle the problem. It successfully yields personalized models based on awareness of similarity between local data, and achieves better tradeoff between generalization and personalization than existing methods. After that, we further design a differentially sparse regularizer to improve communication efficiency during procedure of model training. Additionally, we propose an effective method to reduce the computational cost, which improves computation efficiency significantly. Furthermore, we collect five real medical datasets, including two public medical image datasets and three private multi-center clinical diagnosis datasets, and evaluate its performance by conducting nodule classification, tumor segmentation, and clinical risk prediction tasks. Comparing with 14 existing related methods, the proposed method successfully achieves the best model performance, and meanwhile up to 60% improvement of communication efficiency. Source code is public, and can be accessed at: https://github.com/ApplicationTechnologyOfMedicalBigData/pFedNet-code.**

***Index Terms*—Medical data, federated learning, personalized model, similarity network.**

I. Introduction

With proliferation of data, decision models generated by data-driven paradigm have shown remarkable performance on clinical diagnosis and treatment [1], [2], [3], [4], [5]. Those medical models are usually trained by using multiple institutes' data, which may lead to leakage of privacy due to centralization of medical data. Recently, federated learning has shown significant advantages on alleviating such concerns, since it does not require exchange of medical data between hospitals [6], [7], [8], [9], [10], [11]. More and more federated models have been developed for clinical diagnosis and treatment [12], [13], [14], [15], [16].

Although federated learning has drawn much attention due to its superiority in privacy protection, its performance may face with degradation due to heterogenous data of different medical institutes [17]. For example, as one of data heterogeneity, label unbalance widely exists between comprehensive hospital and specialized hospital, e.g. tumor hospital, which may highly impair model's performance [18]. To mitigate such drawback of federated learning, personalized models are extensively investigated [19], and extensive personalized methods such as FedAMP [20], FedRoD [21], APFL [22], FPFC [23], IFCA [24], pFedMe [25], SuPerFed [26], FedRep [27] have been proposed. Although personalized models yielded by those methods have shown adaption to heterogenous data, they usually have three major limitations in medical scenario, including *sub-optimal performance*, *requirement of prior assumption*, and *limited flexibility*. Specifically, in terms of *sub-optimal performance*, those methods usually work well in some general datasets such as MNIST[1], and CIFAR[2], but have unsatisfied performance in real medical scenario due to high complication of medicine [22], [24], [25]. Additionally, in terms of *requirement of prior assumption*, existing methods may assume either clustering structure among clients [24] or client's computing resources [27], which may be either hard to know or not satisfied in real medical scenario. Moreover, in terms of *limited flexibility*, some existing methods develop personalized models based on similarity network for clients' local data, but limit to few special topologies of network such as complete graph [23], [20], star graph [28], [26], [21], and may not achieve optimum for general medical scenarios directly.

Moreover, another drawback of those existing personalized models is the limited usability for medical application. One of major reasons is that they are not able to handle both shared and personalized components of medical data discriminately. Those components widely exist in clinical diagnosis and treatment [29].

- **Shared component.** Generally, clinical diagnosis and treatment are conducted by referring to international clinical guidelines such as the Clinical Practice Guidelines (CPG)[3], which usually provide general clinical treatment for the whole population. Those clinical guidelines provide some necessary operations for some symptom, no matter who he/she is. For example, when inflammation appears, patients have to conduct blood routine examination before clinical diagnosis. It is widely applied for all patients who appear the corresponding symptom.

Yawei Zhao, Qinghe Liu, Pan Liu, and Kunlun He are with Medical Innovation Research Division, Chinese PLA General Hospital, Beijing, 100835, China. Xinwang Liu is with the School of Computer, National University of Defense Technology, Changsha, Hunan, China. E-mail: csyawei.zhao@gmail.com, liuqinghe9638@163.com, panliu@icloud.com, xinwangliu@nudt.edu.cn, and kunlunhe@plagh.org. † means equal contribution. ‡ represents corresponding author.

[1] http://yann.lecun.com/exdb/mnist/
[2] https://www.cs.toronto.edu/~kriz/cifar.html
[3] https://www.nccih.nih.gov/health/providers/clinicalpractice

- **Personalized component.** Every patient has his/her family genetic history, allergen, medication records etc. Such patient's information is unique, and should be considered carefully before offering clinical diagnosis and treatment. For example, when inflammation appears, the patient who has history of penicillin allergy cannot be treated by using penicillin.

Therefore, it is necessary for personalized models to capture above common and personalized characteristics of medical data. Unfortunately, few existing methods are designed to handle the case. Although Collins et al. develop a local model for every client with shared representation among clients [27], it ignores the underlying similarity between clients who own similar data, and cannot allow to adjust personalization as need for medical scenario.

To mitigate limitations of those existing methods, we propose a new formulation of personalized federated learning, namely pFedNet, and develop a flexible framework to obtain good adaption to heterogenous medical data. The personalized model[4] consists of the shared component and the personalized component, and is designed to capture both common and personalized characteristics of medical data. Note that pFedNet builds personalized models based on the similarity network of clients' data, which is able to find underlying relation between personalized models. Additionally, it does not rely on any extra assumption on clients' clustering structure, and any special topology of similarity network, and thus is more suitable to real medical scenario.

Furthermore, we propose a new communication efficient regularizer to reduce workload of communication between clients and server, which can encourage elements of local update to own clustering structure, and thus improve communication efficiency. After that, we propose a new framework to optimize and obtain personalized models, which successfully reduces computational cost significantly. Finally, we collect five real medical datasets, which includes two public datasets of medical image and three private datasets of medical records. Three classic medical tasks, including nodule classification, tumor segmentation, and clinical risk prediction, are conducted to evaluate the proposed method. Numerical results show that the proposed method successfully outperforms existing methods on performance, and meanwhile achieves up to 60% promotion of communication efficiency. In summary, contributions of the paper are summarized as follows.

- We propose a new personalized federated model for the medical scenario, which is built based on awareness of similarity of between medical institutes' data, and successfully captures both shared and personalized characteristics of patients' data.
- We develop a new communication efficient regularizer to reduce workload of communication during learning of personalized model, and a new optimization framework to reduce the computational cost.
- Extensive empirical studies have been conducted to evaluate the effectiveness of the proposed model and the optimization framework.

The paper is organized as follows. Section II reviews related literatures. Section III presents the proposed formulation, and explains its application. Section IV presents a communication efficient regularizer, which is able to decrease communication's workload during learning of models. Section V presents an efficient method to reduce computation cost during federated learning. Section VI presents extensive empirical studies, and Section VII concludes the paper.

II. Related Work

In the section, we review related literatures on methodology of personalized federated learning, and medical applications of federated learning.

A. Personalized Federated Learning

Personalized federated learning combines benefits of personalized model and federated learning, while taking into account the unique characteristics and preferences of each client [19]. Its methodology usually has five branches including *parameter decoupling*, *knowledge distillation*, *multi-task learning*, *model interpolation*, and *clustering*. Specifically, the branch of *parameter decoupling* classifies parameters of model into two categories: base parameters and personalized parameters, where base parameters are shared between client and server, and personalized parameters are stored at client privately [30], [31], [32]. The branch of *knowledge distillation* transfers the knowledge from teacher's model to student's model, which can significantly enhance the performance of local models [33], [34], [35], [36], [37]. The branch of *multi-task learning* views client's model as a task, and abstracts the learning procedure of personalized federated models as a multi-task learning task [38], [39], [20], [40]. The branch of *model interpolation* simultaneously learns a global model for all clients, and a local model for every client. It usually makes tradeoff between the global model and local models to achieves the optimum of personalization [28], [22], [41]. The branch of *clustering* aims to generating similar personalized model for clients who own similar data distribution [42], [43], [24], [44]. The proposed method of personalized federated learning, namely pFedNet, belongs to the branch of *clustering*, but meanwhile allows to decouple parameters flexibly. Those existing methods in the branch of *clustering* either conduct model learning and client clustering separately [42], [43], or conduct model learning based on prior assumption on clustering, e.g. selection of the number of clusters and specific clustering method [45], [46]. Comparing with them, pFedNet focuses on learning of personalized models, and meanwhile finds clustering structure among clients implicitly. It does not rely on prior assumption on clustering, and thus usually obtain better performance benefiting from good adaption of local data.

Most related methods include two groups: personalized models based on similarity and personalized models with mixture of components. Specifically, the first group consists of methods such as FPFC [23], FedAMP [20], L2GD [28], FedRoD [21], SuPerFed [26] etc. These existing methods

[4]In the paper, we do not distinguish difference between the proposed model and formulation, and denote them by using pFedNet indiscriminately.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3470072

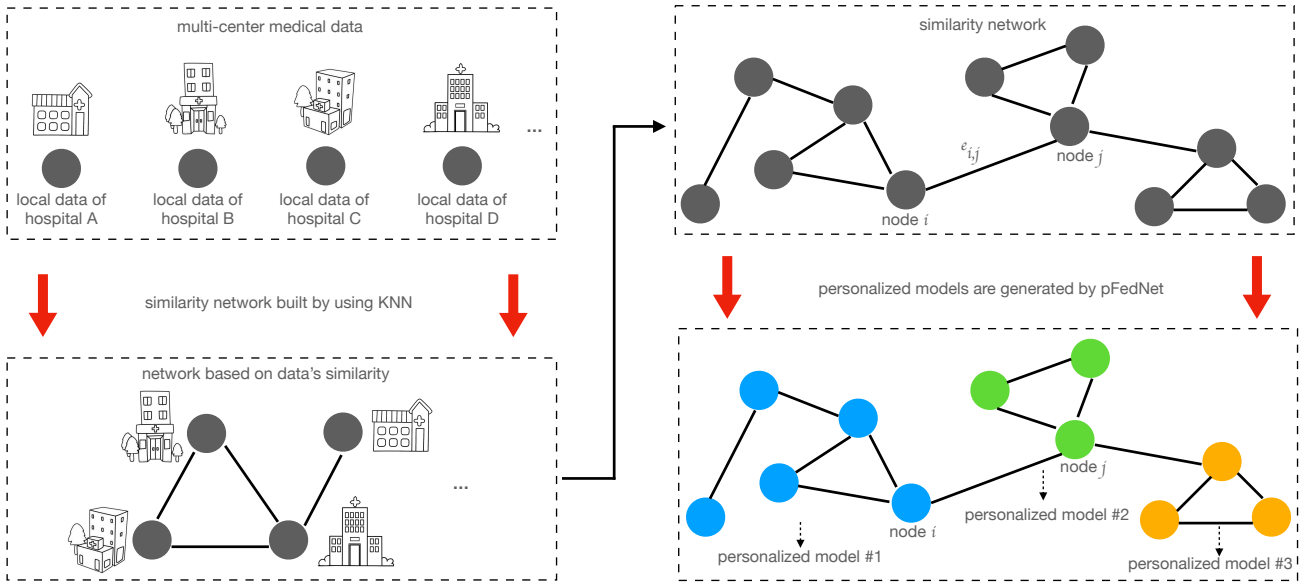


Fig. 1. Personalized models are produced based on similarity of multi-center medical data. As shown in left subfigures, the black node represents a hospital and its local data. The similarity between hospitals' data is measured by using KNN. The corresponding nodes with high similarity are connected by an edge. As shown in right subfigures, given a similarity network, the proposed method pFedNet is able to produce personalized models, where nodes owning similar distribution of data have similar personalized models. Those nodes owning significantly different distribution of data have different models.

yield personalized model based on some special topologies of similarity network of clients' local data, e.g. complete graph and star graph, limiting their applications in medical scenarios. For example, both FPFC and FedAMP generate personalized model based on the complete graph. L2GD, FedRoD, and SuPerFed produce personalized models based on the star graph. Comparing with those methods, the proposed method, namely pFedNet, does not have this limitation, and can work on any topology. Moreover, the second group consists of methods such as FedRep [27], who produce personalized model with mixture of components. However, FedRep assumes every client owns unique local model, and does not consider the similarity between client's data. It pays much attention on the difference between client's model, and ignores similarity between them. Comparing with FedRep, the proposed method supports flexible combination of personalized and shared components, and meanwhile achieves better performance based on awareness of similarity between clients' local data. In fact, FedRep can be viewed as a special case of pFedNet.

B. Federated Learning in Medical Applications

In recent years, several studies have explored the use of federated learning in medicine, and present promising results [47], [48]. One of the main medical applications is the development of predictive models for disease diagnosis and treatment [16], [15]. For example, Bai et al. propose an open source framework for medical artificial intelligence, and offer diagnosis of COVID-19 by using federated learning method [16]. Dayan et al. develop federated learning method to predict clinical outcomes in patients with COVID-19. Additionally, another area where federated learning has shown promising is analysis of medical images [49], [50]. For instance, Kaissis et al. review recent emerging methods on privacy preservation of medical images analysis, and discusses drawbacks and limitations of those existing methods [49]. Moreover, a general federated learning framework, namely PriMIA [48] is developed, and its advantages on privacy protection, securely aggregation, and encrypted inference have been evaluated by conducting classification of paediatric chest X-rays images. Similarly, a federated learning method for predicting histological response to neoadjuvant chemotherapy in triple-negative breast cancer is recently proposed [12]. An automatic tumor boundary detector for the rare disease of glioblastoma has been proposed by using federated learning [13], which presents impressive performance. Similar to those studies, the paper focuses on medical scenario, but provides a general and flexible learning framework for personalized models. The proposed formulation is inspired by the real procedure of clinical treatment, has wide applications for disease diagnosis and medical data analysis, and is not limited to a specific disease like those existing methods.

III. Formulation

In the section, we first present similarity network to represent heterogenous clients' data, and then develop a new formulation of personalized federated learning. Finally, we present the framework of alternative optimization to solve the proposed formulation.

A. Similarity Network for Personalized Representation

In the paper, we use the concept of *network* to measure similarity of local data, and build a similarity network for the proposed formulation. As illustrated in Figure 1, every hospital and its local data is represented by a node. Similarity between nodes' local data is measured by using representative samples which are generated by randomly sampling from local dataset. It is computed by using the distance between those representative samples. When the similarity is significant, those

corresponding nodes are connected by an edge. In the paper, we use K-Nearest Neighbors (KNN) method to find every node's top-k most similar nodes as neighbors. As illustrated in Figure 1, nodes i and j are neighbors, and the element of the i-th row (column) and j-th column (row) of the adjacency matrix is 1. Finally, the similarity network $\mathcal{G} := \{\mathcal{N}, \mathcal{E}\}$ is built, where $\mathcal{N} := \{1, 2, ..., N\}$ represents the node set, consisting of N nodes. $\mathcal{E} := \{e_{i,j} : i \in \mathcal{N}, j \text{ is the node } i\text{'s neighbor}\}$ represents the edge set, consisting of M edges.

Besides, major notations used in the paper are summarized as follows for easy understanding of mathematical details.

- Bold and lower letters such as $\mathbf{a}$ represent a vector. Bold and upper letters such as $\mathbf{X}$ represents a matrix.
- Lower letters such as $f(\cdot)$ and $h(\cdot)$ represent a function. Other letters such as n, N, M represents a scalar value.
- $\mathcal{N}$ and $\mathcal{E}$ represent a set, and $\mathcal{D}_n$ represent a data distribution for the n-th client.
- $\odot$ represents Hadamard of two matrices. $\|\cdot\|_p$ represent the p-th norm of a vector.
- ∇ represents the gradient operator, and $\nabla f(\cdot)$ represent the gradient of f.
- $[\mathbf{a}]_+$ means negative elements of $\mathbf{a}$ is replaced by 0, and non-negative elements of $\mathbf{a}$ do not make any change.

B. pFedNet: Formulation of Personalized Federated Learning

Given the similarity network $\mathcal{G}$, and constant matrices $\mathbf{M} \in \mathbb{R}^{d\times d_1}$ and $\mathbf{N} \in \mathbb{R}^{d\times d_2}$, the proposed personalized federated learning, namely pFedNet, is finally formulated by

$$\min_{\mathbf{x},\{\mathbf{x}^{(n)},\mathbf{z}^{(n)}\}_{n=1}^N} \frac{1}{N}\sum_{n\in\mathcal{N}} f_n\left(\mathbf{x}^{(n)};\mathcal{D}_n\right) + \lambda \sum_{\substack{e_{i,j}\in\mathcal{E},\\ \forall i,j\in\mathcal{N}}} \left\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\right\|_p,$$

subject to:

$$\mathbf{x}^{(n)} = \mathbf{M}\mathbf{x} + \mathbf{N}\mathbf{z}^{(n)}, \qquad \forall n \in \mathcal{N}, \mathbf{x} \in \mathbb{R}^{d_1}, \mathbf{z}^{(n)} \in \mathbb{R}^{d_2}.$$

Here, d_1 and d_2 are scalars, representing number of features of shared and personalized components, respectively. $f_n\left(\mathbf{x}^{(n)};\mathcal{D}_n\right)$ represents the local loss at the n-th client, where $\mathbf{x}^{(n)}$ represents the personalized model, and $\mathcal{D}_n$ represents the local data. For example, it can be instantiated by $f_n\left(\mathbf{x}^{(n)};\mathcal{D}_n\right) = \sum_{(\mathbf{a},y)\sim\mathcal{D}_n} \log\left(\frac{1}{1+e^{-y\mathbf{a}^\top\mathbf{x}^{(n)}}}\right)$ for the logistic regression task, where $(\mathbf{a}, y) \sim \mathcal{D}_n$ represents that an instance $\mathbf{a}$ and its label y are drawn from local dataset $\mathcal{D}_n$.

Note that $\mathbf{x}$ *and* $\mathbf{z}^{(n)}$ *represent the shared and personalized component of the personalized model* $\mathbf{x}^{(n)}$*, respectively.* Those shared and personalized components are priors, and usually determined in practical application scenarios. For example, in terms of statistical machine learning models like SVM and logistic regression [51], their shared component may be weights of features like inflammation, diarrhoea, and vomiting etc, and their personalized component may be weights of features like family genetic history and allergen etc. In terms of deep learning models like dense net [52] and u-net [53], their shared component may be weights of layers of feature extraction, and their personalized component may be weights of layers of classifier.

The proposed formulation has wide application in medical analysis, clinical diagnosis, and treatment. Generally, doctors offer diagnosis and treatment service according to patients' medical records and the international clinical guidelines. It is a natural scenario for personalized model with mixture of components to conduct clinical decision.

- **Personalized component**. Since every patient has his/her unique medical record including family genetic history, allergen, medication records etc, the personalized component of model, e.g. $\mathbf{z}^{(n)}$ is necessary to capture characteristics of such data.
- **Shared component**. The international clinical guideline usually provide a general solution to conduct diagnosis and treatment. For example, blood routine examination is required when inflammation appears. The shared component of model, e.g. $\mathbf{x}$ is necessary to capture such common characteristics.

Additionally, some special diseases such as regional disease, and occupational disease also need personalized model with shared component to conduct clinical decision [29]. Specifically, the treatment of regional and occupational disease needs to consider the location and occupation of patients, respectively, which corresponds to the personalized component of the clinical decision model. Besides, all patients should also be offered some basic treatment such as alleviation of inflammation, which corresponds the shared component of the model. In a nutshell, the formulation provides a general and flexible framework to conduct personalized federated learning.

- **Generality**. No matter statistical machine learning models such as ridge regression, logistic regression, support vector machine etc [51] or deep learning models such as dense net [52] and u-net [53] etc, the formulation can be instantiated by specifying the local loss function f_n.
- **Flexibility**. It is flexible to make tradeoff between personalized need and global requirement based on similarity of local data. Nodes who have similar distribution of data own similar or even identical personalized models. It is mainly manifested in two aspects including the similarity network $\mathcal{G}$ and the hyper-parameter λ. Existing related researches usually require $\mathcal{G}$ to own special topology [23], [20], [28], [26]. However, the proposed formulation, namely pFedNet, does not have this limitation, and can work for any a similarity network.

As we have shown, the similarity network $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ is constructed by using KNN method. The number of neighbours can be set from 1 to $n-1$ when using KNN method. Note that the similarity network becomes dense fast with the increase of the number of neighbours, and thus leads to rapid growth of computational cost. We recommend to construct the similarity network by choosing a small number, for example $k = 3$, which has been proved to have superiority of performance against existing methods. Additionally, λ with $\lambda > 0$ is a given hyper-parameter, which controls the personalization of federated model. When $\lambda \to 0$, more personalization is allowed. The personalization decays with the increase of λ. Almost all $\mathbf{z}^{(n)}$ with $n \in \{1, 2, \cdots, N\}$ tend to be same for a large λ.

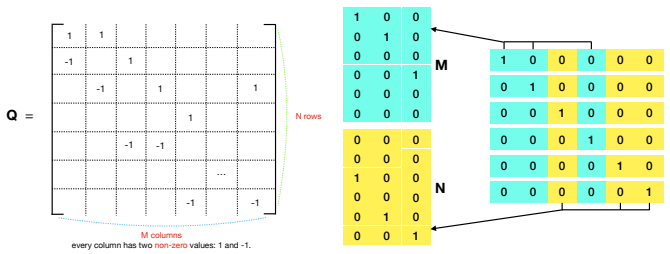


Fig. 2. Every column of $\mathbf{Q}$ has two non-zero elements: 1 and -1. Every column of both $\mathbf{M}$ and $\mathbf{N}$ has one non-zero element: 1.

C. Optimization

Note that the formulation can be equally transformed as follows.

$$\min_{\{\mathbf{z}^{(n)}\}_{n=1}^{N},\mathbf{x}} \frac{1}{N}\sum_{n=1}^{N} f_n\left(\mathbf{x}^{(n)};\mathcal{D}_n\right)+\lambda\left\|\mathbf{ZQ}\right\|_{1,p}, \quad (1)$$

subject to:

$$\mathbf{x}^{(n)}=\mathbf{Mx}+\mathbf{Nz}^{(n)}.$$

Here, $\mathbf{Z}\in\mathbb{R}^{d_2\times N}$, and $\mathbf{Q}\in\mathbb{R}^{N\times M}$. Denote the n-th column of $\mathbf{Z}$ by $\mathbf{z}^{(n)}\in\mathbb{R}^{d_2}$, that is $\mathbf{Z}=\left[\mathbf{z}^{(1)},\mathbf{z}^{(2)},...,\mathbf{z}^{(N)}\right]$. N and M represent the total number of nodes and edges in the network $\mathcal{G}$, respectively. As shown in Figure 2, both $\mathbf{M}$ and $\mathbf{N}$ have special structure, where every row of them has at most one non-zero value, and the non-zero value is 1. $p\in\{1,2,\infty\}$. $\mathbf{Q}$ is the given auxiliary matrix, which has M columns and every column has two non-zero values: 1 and -1. Note that $\|\cdot\|_{1,p}$ is denoted by $\ell_{1,p}$ norm. Given a matrix $\mathbf{U}\in\mathbb{R}^{d_2\times M}$, it is defined by

$$\left\|\mathbf{U}\right\|_{1,p}:=\sum_{m=1}^{M}\left\|\mathbf{U}_{:,m}\right\|_p.$$

Formulation 1 is difficult to solve for three reasons. First, the optimization variables may be highly non-separable due to $\mathbf{Q}$. As we have shown, every column of $\mathbf{Q}$ corresponds an edge of the similarity network $\mathcal{G}$, which implies that the corresponding personalized component, e.g. $\mathbf{z}^{(i)}$ and $\mathbf{z}^{(j)}$, corresponding to nodes of such edge has dependent relation. Second, the loss function may be highly non-smooth, because the regularizer is sum of norms. Third, the number of optimization variables is large, when the network $\mathcal{G}$ has a large number of nodes and edges. Generally, Formulation 1 is solved by alternative optimization. The variable $\{\mathbf{x}^{(1)},\mathbf{x}^{(2)},...,\mathbf{x}^{(N)}\}$ is obtained by alternatively optimizing $\mathbf{x}$ and $\{\mathbf{z}^{(1)},\mathbf{z}^{(2)},...,\mathbf{z}^{(N)}\}$.

Optimizing x by given Z. $\mathbf{x}$ is optimized by:

$$\min_{\mathbf{x}\in\mathbb{R}^{d_1}}\frac{1}{N}\sum_{n=1}^{N} f_n\left(\mathbf{Mx}+\mathbf{Nz}^{(n)};\mathcal{D}_n\right).$$

By using the data-driven stochastic optimization method such as SGD [54], we update $\mathbf{x}$ by solving

$$\min_{\mathbf{x}\in\mathbb{R}^{d_1}}\frac{1}{N}\sum_{n=1}^{N}\left\langle\mathbf{M}^{\top}\mathbf{g}_t^{(n)},\mathbf{x}\right\rangle+\frac{1}{2\eta_t}\left\|\mathbf{x}-\mathbf{x}_t\right\|^2, \quad (2)$$

Algorithm 1 Compute local stochastic gradient at the n-th client for the t+1-th iteration.

1: Receive the personalized model $\mathbf{y}_t^{(n)}:=\mathbf{Mx}_t+\mathbf{Nz}_t^{(n)}$ from the server.
2: Randomly sample an instance $\mathbf{a}\sim\mathcal{D}_n$, and compute the stochastic gradient $\mathbf{g}_t^{(n)}=\nabla f(\mathbf{y}_t^{(n)};\mathbf{a})$ with $\mathbf{a}\sim\mathcal{D}_n$.
3: Send $\mathbf{g}_t^{(n)}$ to the server.

Algorithm 2 Train personalized models at the server.

Require: The number of total iterations T, and the initial model $\mathbf{x}_1$, $\mathbf{z}_1^{(n)}$ with $n\in\{1,2,\cdots,N\}$.
1: Deliver the model $\mathbf{y}_1^{(n)}=\mathbf{Mx}_1+\mathbf{Nz}_1^{(n)}$ to all client n with $n\in\{1,2,...,N\}$.
2: **for** $t=1,2,...,T$ **do**
3: Collect stochastic gradient $\mathbf{G}_t=\left[\mathbf{g}_t^{(1)},\mathbf{g}_t^{(2)},...,\mathbf{g}_t^{(N)}\right]$ from all client n with $n\in\{1,2,...,N\}$.
4: Update the global model $\mathbf{x}$ by solving 2.
5: Update the personalized model $\mathbf{Z}$ by solving 3.
6: Deliver the parameter $\mathbf{y}_{t+1}^{(n)}=\mathbf{Mx}_{t+1}+\mathbf{Nz}_{t+1}^{(n)}$ to every client.
return $\mathbf{x}_{T+1}^{(n)}=\mathbf{Mx}_{T+1}+\mathbf{Nz}_{T+1}^{(n)}$ with $n\in\{1,2,...,N\}$.

where $\mathbf{g}_t^{(n)}$ is a stochastic gradient of f_n with $\mathbf{Mx}+\mathbf{Nz}_t^{(n)}$ by using data drawn from the local dataset $\mathcal{D}_n$.

Optimizing Z by given x. $\mathbf{Z}$ is optimized by:

$$\min_{\mathbf{Z}\in\mathbb{R}^{d_2\times N}}\frac{1}{N}\sum_{n=1}^{N} f_n\left(\mathbf{Mx}+\mathbf{Nz}^{(n)};\mathcal{D}_n\right)+\lambda\left\|\mathbf{ZQ}\right\|_{1,p}.$$

By using the data-driven stochastic optimization method such as SGD [54], we need to solve:

$$\min_{\mathbf{Z}\in\mathbb{R}^{d_2\times N}}\frac{1}{N}\sum_{n=1}^{N}\left\langle\mathbf{N}^{\top}\mathbf{g}_t^{(n)},\mathbf{z}^{(n)}\right\rangle+\lambda\left\|\mathbf{ZQ}\right\|_{1,p}+\frac{\left\|\mathbf{Z}-\mathbf{Z}_t\right\|_F^2}{2\eta_t}.$$

$\mathbf{g}_t^{(n)}$ is a stochastic gradient of f_n with $\mathbf{Mx}_t+\mathbf{Nz}^{(n)}$ by using stochastic data drawn from the local dataset $\mathcal{D}_n$. Suppose $\mathbf{G}_t=\left[\mathbf{g}_t^{(1)},\mathbf{g}_t^{(2)},...,\mathbf{g}_t^{(N)}\right]$, and $\mathbf{Z}$ is optimized by solving:

$$\min_{\mathbf{Z}\in\mathbb{R}^{d_2\times N}}\frac{\mathbf{1}_d^{\top}\left(\left(\mathbf{N}^{\top}\mathbf{G}_t\right)\odot\mathbf{Z}\right)\mathbf{1}_N}{N}+\lambda\left\|\mathbf{ZQ}\right\|_{1,p}+\frac{\left\|\mathbf{Z}-\mathbf{Z}_t\right\|_F^2}{2\eta_t}. \quad (3)$$

Here, $\odot$ means Hadamard product of two matrices.

Federated optimization. According to the above optimization steps, the stochastic gradient $\mathbf{G}_t$ is obtained at client in the scenario of federated learning. Algorithm 1 shows that every client receives its personalized model from server, and returns update of parameters by using local data. Algorithm 2 demonstrates that the server receives local update of parameters, computes the global and personalized models, and finally returns those personalized models to clients. Unfortunately, the federated optimization has two major drawbacks.

- **Heavy workload of communication**. Since every client has to transmit the stochastic gradient, e.g. $\mathbf{g}_t^{(n)}$ to the server, the communication workload will be unbearable

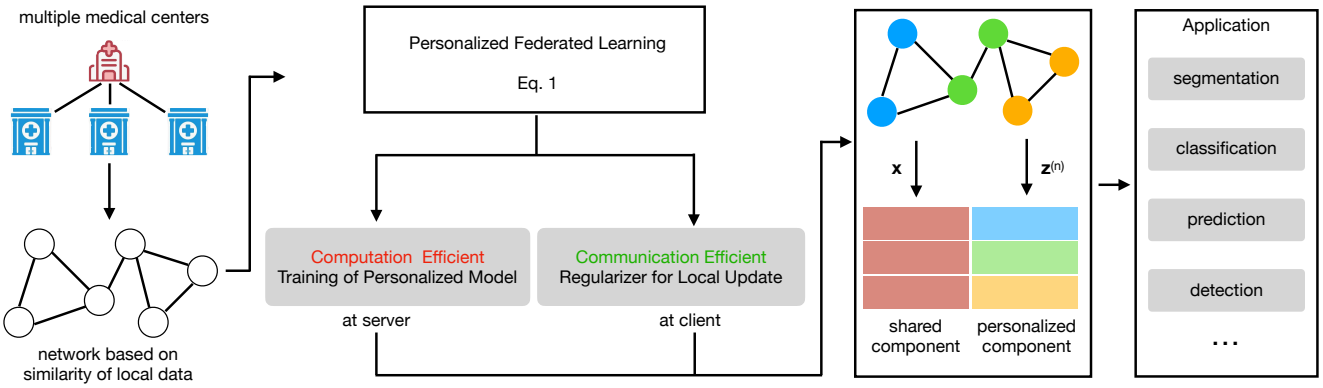


Fig. 3. Federated model with mixture of components is formulated based on similarity network of distributed data, and then trained by communication-efficient update of parameters at client and computation-efficient aggregation of parameters at server.

for a large d. Especially, deep neural network models usually own more than millions of parameters, the transmission of such gradient will lead to high cost of communication.

- **High cost of computation**. Since the sum-of-norms regularizer leads to high non-separability and non-smoothness of the objective loss, the computation cost is high. The optimization of personalized model is time-consuming and even unbearable.

To mitigate those drawbacks, we first develop a communication efficient method for every client to transmit the stochastic gradient, which will be presented in Section IV. Additionally, we propose a computation efficient method for the server to update the personalized model, which will be presented in Section V. In summary, the framework of personalized federated learning is illustrated in Figure 3.

IV. Communication Efficient Update of Model

In the section, we first propose a communication efficient regularizer, which encourages elements of update of local model to own clustering structure, and improves the communication efficiency effectively. Then, we develop an ADMM method [55] to conduct the update of local model.

A. CER: Communication Efficient Regularizer

Recall that traditional federated learning methods such as Algorithm 1 have to send local update $\mathbf{g}_t^{(n)}$ to the server. It may consume a large amount of bandwidth, and lead to high communication cost. To improve the communication efficiency, we propose a communication efficient method, which can let $\mathbf{g}_t^{(n)}$ be encoded by using few bits. Since the code length of $\mathbf{g}_t^{(n)}$ is reduced, the communication efficiency is significantly promoted.

Specifically, suppose the update of model is denoted by $\nabla_{t+1}^{(n)}$ for the n-th client at the t-th iteration. Given the local model $\mathbf{y}_t^{(n)}$, it is defined by

$$\nabla_{t+1}^{(n)} := \frac{\mathbf{y}_t^{(n)} - \mathbf{v}}{\eta_t}.$$

Here, $\mathbf{v}$ is an auxiliary variable which is defined by

$$\mathbf{v} := \underset{\mathbf{y}\in\mathbb{R}^d}{\operatorname{argmin}} \left\langle \mathbf{g}_t^{(n)}, \mathbf{y} \right\rangle + \underbrace{\gamma \left\| \mathbf{\Lambda} \left(\mathbf{y} - \mathbf{y}_t^{(n)} \right) \right\|_1}_{\text{communication efficient regularizer}} + \frac{\left\| \mathbf{y} - \mathbf{y}_t^{(n)} \right\|^2}{2\eta_t}. \tag{4}$$

The given full rank square matrix $\mathbf{\Lambda} \in \mathbb{R}^{d\times d}$ is denoted by

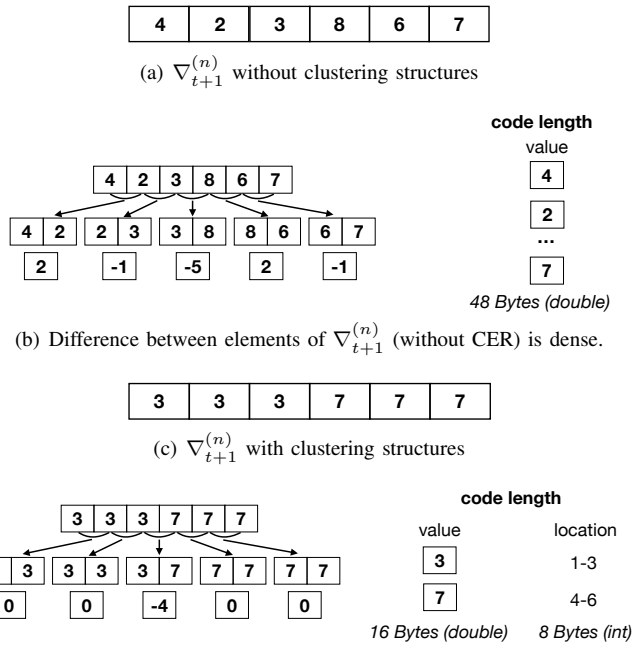
$$\mathbf{\Lambda} = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \cdots & & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix}.$$

Notice that $\mathbf{\Lambda}$ is a full rank square matrix, whose smallest singular value (denoted by σ) is positive, that is $\sigma > 0$.

The basic idea is to induce the clustering structure of elements of $\mathbf{g}_t^{(n)}$ by using differential sparsity regularizer (denoted by CER). The regularizer encourages the update of local model, that is $\nabla_{t+1}^{(n)}$, to own clustering structures. Recall the definition of $\mathbf{v}$, namely Eq. 4, and we observe that CER punishes the difference between elements of $\nabla_{t+1}^{(n)}$, and encourages them to be small or even zero. Thus, those corresponding elements of $\nabla_{t+1}^{(n)}$ are very similar or even identical. That is, the elements of $\nabla_{t+1}^{(n)}$ own clustering structures. Exploiting the clustering structures, $\nabla_{t+1}^{(n)}$ can be compressed by using few bits, and thus improves the communication efficiency in the distributed setting.

Figure 4 presents an illustrative example. According to Figures 4(a) and 4(c), when the elements of $\nabla_{t+1}^{(n)}$ own clustering structures, they can be encoded by using fewer bits. Its code length can be reduced a lot. The update of parameter can be transmitted from clients and the server efficiently. According to Figures 4(b) and 4(d), our basic idea is to let the difference between the elements of $\nabla_{t+1}^{(n)}$ be sparse, which encourages the elements of $\nabla_{t+1}^{(n)}$ to have clustering structures. Comparing with the gradient quantization methods in the previous studies, the proposed method is able to find a good tradeoff between the convergence performance and the communication efficiency.

We present more explanations by taking an example. As illustrated in Figure 5, we generate local update of the person-



(a) $\nabla_{t+1}^{(n)}$ without clustering structures

(b) Difference between elements of $\nabla_{t+1}^{(n)}$ (without CER) is dense.

(c) $\nabla_{t+1}^{(n)}$ with clustering structures

(d) Difference between elements of $\nabla_{t+1}^{(n)}$ (with CER) is **sparse**.

Fig. 4. The illustrative example shows that $\nabla_{t+1}^{(n)}$ with clustering structures can be compressed by using fewer bits, and thus the code length is reduced effectively.

alized model with 100 features (orange lines in Figures 5(e)-5(h)) and difference of its elements (orange lines in Figures 5(a)-5(d)). As we can see, the differential sparsity, e.g. $\mathbf{\Lambda}\nabla_{t+1}^{(n)}$ (blue lines in Figures 5(a)-5(d)) becomes sparse significantly with the increase of γ (Figures 5(a)-5(d)). It verifies that the proposed method, namely CER successfully encourages difference between elements of local update to be sparse. Meanwhile, we find that $\nabla_{t+1}^{(n)}$ is similar to $\mathbf{g}_t^{(n)}$ for a small γ (Figure 5(e)), and a large γ leads to a significant trend (Figures 5(e)-5(h)). As illustrated in Figures 5(d) and 5(h), we observe that elements of local update become similar when their difference is sparse, and thus appear clustering structures (peak and bottom of the blue curve). It leads to much easier compression than the original local update. Note that there is a trade-off between the accuracy and communication efficiency. When elements of a gradient are partitioned into more clusters, the higher accuracy of the gradient is guaranteed. Meanwhile, the gradient has to be encoded by using more bytes, thus leading to the decrease of the communication efficiency.

Additionally, note that the proposed method, namely CER naturally enjoys benefits of privacy protection. As we have shown, we choose transmit $\nabla_{t+1}^{(n)}$ instead of $\mathbf{g}_t^{(n)}$ to conduct training of shared component of the model. It cannot recover the real shared component of the model even though $\nabla_{t+1}^{(n)}$ is exploited to malignant nodes but without known of $\mathbf{g}_t^{(n)}$.

B. Optimizing $\nabla_{t+1}^{(n)}$

We use ADMM [55] to find the optimum of $\nabla_{t+1}^{(n)}$. As we have shown, $\nabla_{t+1}^{(n)}$ is obtained by

$$\nabla_{t+1}^{(n)} = \frac{\mathbf{x}_t - \mathbf{v}}{\eta_t}.$$

Here, $\mathbf{v}$ can be obtained by performing:

$$\min_{\mathbf{y}\in\mathbb{R}^d, \mathbf{r}\in\mathbb{R}^d} \underbrace{\|\mathbf{r}\|_1 + \frac{\left\|\mathbf{y} - \left(\mathbf{y}_t^{(n)} - \eta_t \mathbf{g}_t^{(n)}\right)\right\|^2}{2\eta_t\gamma}}_{=:g(\mathbf{r},\mathbf{y})},$$

subject to:

$$\mathbf{r} = \mathbf{\Lambda}\mathbf{y} - \mathbf{\Lambda}\mathbf{y}_t^{(n)}.$$

ADMM[55] is used to solve the above optimization problem, which consists of update of $\mathbf{r}$, $\mathbf{y}$, and $\boldsymbol{\omega}$, iteratively. The augmented Lagrangian $L_\rho(\mathbf{r},\mathbf{y},\boldsymbol{\omega})$ is defined by

$$\begin{aligned} &L_\rho(\mathbf{r},\mathbf{y},\boldsymbol{\omega}) \\ :=&g(\mathbf{r},\mathbf{y}) + \left\langle \boldsymbol{\omega}, \mathbf{r} - \mathbf{\Lambda}\mathbf{y} + \mathbf{\Lambda}\mathbf{y}_t^{(n)} \right\rangle + \frac{\rho}{2}\left\|\mathbf{r} - \mathbf{\Lambda}\mathbf{y} + \mathbf{\Lambda}\mathbf{y}_t^{(n)}\right\|^2. \end{aligned}$$

Here, $\mathbf{r}$ and $\mathbf{y}$ are primal variables, and $\boldsymbol{\omega}$ is the dual variable. ρ is called the penalty parameter. Note that the objective function is closed, proper, and convex, and the unaugmented Lagrangian $L_0(\mathbf{r},\mathbf{y},\boldsymbol{\omega})$ (for the case of $\rho = 0$) has a saddle point. Thus, ADMM achieves convergence, and the objective function approaches the optimal value [55].

Update of r. Given $\mathbf{y}_j$ and $\boldsymbol{\omega}_j$, $\mathbf{r}_{j+1}$ is updated by performing:

$$\begin{aligned} &\mathbf{r}_{j+1} = \operatorname*{argmin}_{\mathbf{r}\in\mathbb{R}^d} L_\rho(\mathbf{r},\mathbf{y}_j,\boldsymbol{\omega}_j) \\ =&\operatorname*{argmin}_{\mathbf{r}\in\mathbb{R}^d} \|\mathbf{r}\|_1 + \frac{\rho}{2}\left\|\mathbf{r} - \left(\mathbf{\Lambda}\mathbf{y}_j - \mathbf{\Lambda}\mathbf{y}_t^{(n)} - \frac{1}{\rho}\boldsymbol{\omega}_j\right)\right\|^2 \\ =&\mathbf{Prox}_{\rho,\|\cdot\|_1}\left(\mathbf{\Lambda}\mathbf{y}_j - \mathbf{\Lambda}\mathbf{y}_t^{(n)} - \frac{1}{\rho}\boldsymbol{\omega}_j\right) \\ =&\left[\mathbf{\Lambda}\left(\mathbf{y}_j - \mathbf{y}_t^{(n)}\right) - \frac{\boldsymbol{\omega}_j}{\rho} - \rho\right]_+ - \left[\mathbf{\Lambda}\left(\mathbf{y}_t^{(n)} - \mathbf{y}_j\right) + \frac{\boldsymbol{\omega}_j}{\rho} - \rho\right]_+. \end{aligned} \tag{5}$$

Here, '**Prox**' represents the proximal operator [56], which is defined by

$$\mathbf{Prox}_{\nu,\phi}(\mathbf{a}) := \operatorname*{argmin}_{\mathbf{b}} \phi(\mathbf{b}) + \frac{\nu}{2}\|\mathbf{b} - \mathbf{a}\|^2.$$

The last equality holds due to

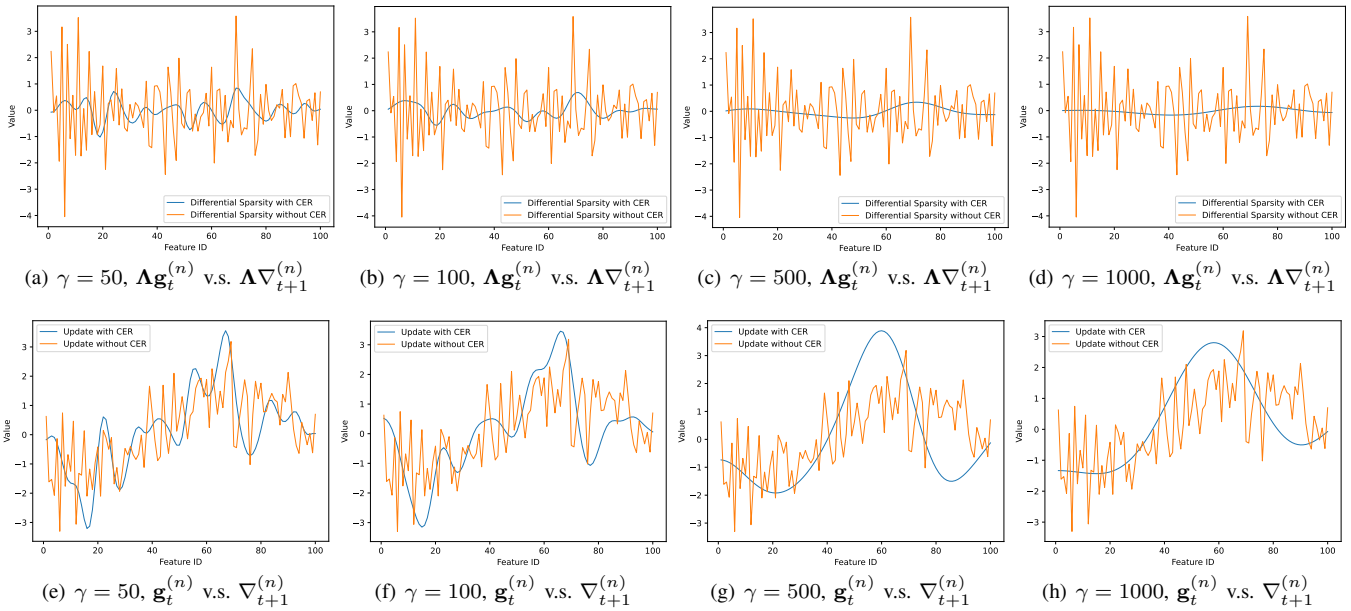
$$\mathbf{Prox}_{\nu,\|\cdot\|_1}(\mathbf{a}) = (\mathbf{a} - \nu)_+ - (-\mathbf{a} - \nu)_+,$$

where $\mathbf{b}_+$ means that negative elements of $\mathbf{b}$ are set by 0, and other non-negative elements do not change.

Update of y. Given $\mathbf{r}_{j+1}$ and $\boldsymbol{\omega}_j$, $\mathbf{y}_{j+1}$ is updated by solving:

$$\begin{aligned} &\mathbf{y}_{j+1} = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^d} L_\rho(\mathbf{r}_{j+1},\mathbf{y},\boldsymbol{\omega}_j) \\ =&\operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^d} \left\|\mathbf{\Lambda}\mathbf{y} - \left[\mathbf{r}_{j+1} + \mathbf{\Lambda}\mathbf{y}_t^{(n)} + \frac{\boldsymbol{\omega}_j}{\rho}\right]\right\|^2 + \frac{\left\|\mathbf{y} - \mathbf{y}_t^{(n)} + \eta_t\mathbf{g}_t^{(n)}\right\|^2}{\rho\eta_t\gamma} \\ =&\left(\rho\eta_t\gamma\mathbf{\Lambda}^\top\mathbf{\Lambda} + \mathbf{I}\right)^{-1}\left[\rho\eta_t\gamma\mathbf{\Lambda}^\top\left[\mathbf{r}_{j+1} + \mathbf{\Lambda}\mathbf{y}_t^{(n)} + \frac{\boldsymbol{\omega}_j}{\rho}\right] + \mathbf{y}_t^{(n)} - \eta_t\mathbf{g}_t^{(n)}\right]. \end{aligned}$$

According to the eigenvalue decomposition, $\mathbf{\Lambda}^\top\mathbf{\Lambda}$ can be represented by $\mathbf{\Lambda}^\top\mathbf{\Lambda} = \mathbf{P}\mathbf{\Sigma}\mathbf{P}^{-1}$, where $\mathbf{\Sigma} :=$



(a) $\gamma = 50$, $\mathbf{\Lambda}\mathbf{g}_t^{(n)}$ v.s. $\mathbf{\Lambda}\nabla_{t+1}^{(n)}$ (b) $\gamma = 100$, $\mathbf{\Lambda}\mathbf{g}_t^{(n)}$ v.s. $\mathbf{\Lambda}\nabla_{t+1}^{(n)}$ (c) $\gamma = 500$, $\mathbf{\Lambda}\mathbf{g}_t^{(n)}$ v.s. $\mathbf{\Lambda}\nabla_{t+1}^{(n)}$ (d) $\gamma = 1000$, $\mathbf{\Lambda}\mathbf{g}_t^{(n)}$ v.s. $\mathbf{\Lambda}\nabla_{t+1}^{(n)}$

(e) $\gamma = 50$, $\mathbf{g}_t^{(n)}$ v.s. $\nabla_{t+1}^{(n)}$ (f) $\gamma = 100$, $\mathbf{g}_t^{(n)}$ v.s. $\nabla_{t+1}^{(n)}$ (g) $\gamma = 500$, $\mathbf{g}_t^{(n)}$ v.s. $\nabla_{t+1}^{(n)}$ (h) $\gamma = 1000$, $\mathbf{g}_t^{(n)}$ v.s. $\nabla_{t+1}^{(n)}$

Fig. 5. Differential sparsity is induced by using CER according to subfigures 5(a)-5(d). Update of model with clustering structure is generated by using CER according to subfigures 5(e)-5(h). $\mathbf{g}_t^{(n)}$ is obtained by setting $\gamma = 0$. $\nabla_{t+1}^{(n)}$ is obtained by setting $\gamma > 0$.

Algorithm 3 Communication efficient update of local models on the n-th client for the $t+1$ iteration.

Require: A positive γ to improve communication efficiency. Given $\mathbf{x}_t$, $\mathbf{P}$, and $\mathbf{\Sigma}$ such that $\mathbf{\Lambda}^\top\mathbf{\Lambda} = \mathbf{P}\mathbf{\Sigma}\mathbf{P}^{-1}$.

1: Receive the personalized model $\mathbf{y}_t^{(n)} := \mathbf{M}\mathbf{x}_t + \mathbf{N}\mathbf{z}_t^{(n)}$ from the server.
2: Randomly sample an instance $\mathbf{a} \sim \mathcal{D}_n$, and compute the stochastic gradient $\mathbf{g}_t^{(n)} = \nabla f(\mathbf{y}_t^{(n)}; \mathbf{a})$ with $\mathbf{a} \sim \mathcal{D}_n$.
3: **for** $j = 0, 1, 2, ..., J-1$ **do**
4: Update $\mathbf{r}_{j+1}$ by performing Eq. 5.
5: Update $\mathbf{y}_{j+1}$ by performing Eq. 6.
6: Update $\boldsymbol{\omega}_{j+1}$ by performing Eq. 7.
7: Compute $\nabla_{t+1}^{(n)}$ with $\nabla_{t+1}^{(n)} = \frac{\mathbf{y}_t^{(n)} - \mathbf{y}_J}{\eta_t}$.
8: Send $\nabla_{t+1}^{(n)}$ to the server.

$\text{diag}\,(\lambda_1, \lambda_2, ..., \lambda_d)$, and λ_i with $i \in \{1, 2, ..., d\}$ are eigenvalues of $\mathbf{\Lambda}^\top\mathbf{\Lambda}$. We have

$$\left(\rho\eta_t\gamma\mathbf{\Lambda}^\top\mathbf{\Lambda} + \mathbf{I}\right)^{-1} = \mathbf{P}\left(\rho\eta_t\gamma\mathbf{\Sigma} + \mathbf{I}\right)^{-1}\mathbf{P}^{-1}.$$

Therefore, $\mathbf{y}_{j+1}$ is updated by performing:

$$\begin{aligned}&\mathbf{y}_{j+1} \\ =&\mathbf{P}\left(\rho\eta_t\gamma\mathbf{\Sigma}+\mathbf{I}\right)^{-1}\mathbf{P}^{-1}\left[\rho\eta_t\gamma\mathbf{\Lambda}^\top\left[\mathbf{r}_{j+1}+\mathbf{\Lambda}\mathbf{y}_t^{(n)}+\frac{\boldsymbol{\omega}_j}{\rho}\right]+\mathbf{y}_t^{(n)}-\eta_t\mathbf{g}_t^{(n)}\right].\end{aligned} \tag{6}$$

Update of $\boldsymbol{\omega}$. Given $\mathbf{r}_{j+1}$ and $\mathbf{y}_{j+1}$, $\boldsymbol{\omega}_{j+1}$ is updated by the following rule:

$$\boldsymbol{\omega}_{j+1} = \boldsymbol{\omega}_j + \rho\left(\mathbf{r}_{j+1} - \mathbf{\Lambda}\mathbf{y}_{j+1} + \mathbf{\Lambda}\mathbf{y}_t^{(n)}\right). \tag{7}$$

Algorithmic details are illustrated in Algorithm 3. Every client receives a personalized model, computes the stochastic gradient by using local data, obtains the communication efficient update of the local model by performing ADMM, and finally sends it to the server. Note that every step of ADMM can be solved by using closed-form solutions, and can be performed efficiently. Update of $\mathbf{r}$ and $\boldsymbol{\omega}$ can be completed by basic matrix operations such as matrix multiplication. Although the update of $\mathbf{y}$ requires to compute the eigenvalue decomposition of $\mathbf{\Lambda}^\top\mathbf{\Lambda}$, it only needs to be performed once, and can be reused repeatedly. Therefore, the update of $\mathbf{y}$ does not lead to high cost of computation. That is to say, the communication efficiency can be improved significantly while it does not lead to high computational cost for every client. In the following section, we furthermore propose a computation efficient method to update the personalized model at the server.

V. Computation Efficient Update of Model

In the section, we find optimum of $\mathbf{x}$ and $\mathbf{Z}$ by performing alternative optimization iteratively.

A. *Efficient update of* $\mathbf{x}$

When the server collects $\nabla_{t+1}^{(n)}$ from client, the shared component $\mathbf{x}$ is updated by performing:

$$\mathbf{x}_{t+1} = \underset{\mathbf{x}\in\mathbb{R}^{d_1}}{\text{argmin}} \left\langle \frac{1}{N}\sum_{n=1}^{N}\mathbf{M}^\top\nabla_{t+1}^{(n)}, \mathbf{x}\right\rangle + \frac{1}{2\eta_t}\left\|\mathbf{x} - \mathbf{x}_t\right\|^2.$$

That is,

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t\left(\frac{1}{N}\sum_{n=1}^{N}\mathbf{M}^\top\nabla_{t+1}^{(n)}\right).$$

As we can see, the shared component $\mathbf{x}$ is updated by multiplication of matrices, which leads to low computational cost.

B. Efficient update of $\mathbf{Z}$

Denote

$$h(\mathbf{Z}) := \frac{\mathbf{1}_d^\top \left(\left(\mathbf{N}^\top \nabla_{t+1}\right) \odot \mathbf{Z}\right) \mathbf{1}_N}{N} + \frac{1}{2\eta_t} \left\|\mathbf{Z} - \mathbf{Z}_t\right\|_F^2 .$$

The update of $\mathbf{Z}$ can be formulated by:

$$\min_{\mathbf{Z}\in\mathbb{R}^{d_2\times N}, \mathbf{W}\in\mathbb{R}^{d_2\times M}} H(\mathbf{Z},\mathbf{W}) := h(\mathbf{Z}) + \lambda \left\|\mathbf{W}\right\|_{1,p},$$

subject to:

$$\mathbf{ZQ} - \mathbf{W} = \mathbf{0}.$$

The augmented Lagrangian $L_\rho(\mathbf{Z},\mathbf{W},\mathbf{\Omega})$ is

$$\begin{aligned} &L_\rho(\mathbf{Z},\mathbf{W},\mathbf{\Omega}) \\ :=&h(\mathbf{Z})+\lambda \left\|\mathbf{W}\right\|_{1,p} + \mathbf{1}_{d_2}^\top(\mathbf{\Omega}\odot(\mathbf{ZQ}-\mathbf{W}))\mathbf{1}_M + \frac{\rho}{2}\left\|\mathbf{W}-\mathbf{ZQ}\right\|_F^2 . \end{aligned}$$

Here, $\mathbf{Z}$ and $\mathbf{W}$ are primal variables, and $\mathbf{\Omega}$ is the dual variable. ρ is called the penalty parameter[5]. Since the objective function is closed, proper, and convex, and the unaugmented Lagrangian $L_0(\mathbf{r},\mathbf{y},\boldsymbol{\omega})$ (for the case of $\rho = 0$) has a saddle point, ADMM is proved to achieve convergence, and the objective function approaches the optimal value [55].

Update of Z. Given $\mathbf{W}_k$ and $\mathbf{\Omega}_k$, $\mathbf{Z}_{k+1}$ is obtained by performing:

$$\begin{aligned} &\mathbf{Z}_{k+1} = \underset{\mathbf{Z}\in\mathbb{R}^{d_2\times N}}{\operatorname{argmin}} L_\rho(\mathbf{Z},\mathbf{W}_k,\mathbf{\Omega}_k) \\ &= \underset{\mathbf{Z}\in\mathbb{R}^{d_2\times N}}{\operatorname{argmin}} h(\mathbf{Z}) + \mathbf{1}_{d_2}^\top(\mathbf{\Omega}_k\odot(\mathbf{ZQ}))\mathbf{1}_M + \frac{\rho}{2}\left\|\mathbf{W}_k-\mathbf{ZQ}\right\|_F^2 . \end{aligned}$$

It is equal to:

$$\begin{aligned} &\mathbf{Z}_{k+1} \\ &= \left[\eta_t\left[\rho\mathbf{W}_k\mathbf{Q}^\top - \mathbf{\Omega}_k\mathbf{Q}^\top - \frac{\mathbf{N}^\top\nabla_{t+1}}{N}\right] + \mathbf{Z}_t\right]\left(\mathbf{I}_N + \eta_t\rho\mathbf{QQ}^\top\right)^{-1} . \end{aligned} \tag{8}$$

Update of W. Given $\mathbf{Z}_{k+1}$ and $\mathbf{\Omega}_k$, $\mathbf{W}_{k+1}$ is obtained by solving:

$$\begin{aligned} &\mathbf{W}_{k+1} = \underset{\mathbf{W}\in\mathbb{R}^{d_2\times M}}{\operatorname{argmin}} L_\rho(\mathbf{Z}_{k+1},\mathbf{W},\mathbf{\Omega}_k) \\ &= \underset{\mathbf{W}\in\mathbb{R}^{d_2\times M}}{\operatorname{argmin}} \lambda\left\|\mathbf{W}\right\|_{1,p} + \frac{\rho}{2}\left\|\mathbf{W} - \left(\mathbf{Z}_{k+1}\mathbf{Q} + \frac{1}{\rho}\mathbf{\Omega}_k\right)\right\|_F^2 \\ &= \mathbf{Prox}_{\frac{\rho}{\lambda},\|\cdot\|_{1,p}}\left(\mathbf{Z}_{k+1}\mathbf{Q} + \frac{1}{\rho}\mathbf{\Omega}_k\right). \end{aligned}$$

Recall that $\left\|\cdot\right\|_{1,p}$ is the sum of norms, its proximal operator has a closed-form [56]. Specifically, the m-th column with $m \in \{1,2,...,M\}$ of $\mathbf{W}_{k+1}$ is obtained by performing:

$$\begin{aligned} &[\mathbf{W}_{k+1}]_{:,m} = \left[\mathbf{Prox}_{\frac{\rho}{\lambda},\|\cdot\|_{1,p}}\left(\mathbf{Z}_{k+1}\mathbf{Q} + \frac{1}{\rho}\mathbf{\Omega}_k\right)\right]_{:,m} \\ &= \left[1 - \frac{\lambda}{\left\|\rho\mathbf{Z}_{k+1}\mathbf{Q}_{:,m} + [\mathbf{\Omega}_k]{:,m}\right\|_q}\right]_+ \left(\mathbf{Z}_{k+1}\mathbf{Q}_{:,m} + \frac{[\mathbf{\Omega}_k]_{:,m}}{\rho}\right), \end{aligned} \tag{9}$$

where $[\mathbf{A}]_{:,m}$ represents the m-th column of $\mathbf{A}$, and $\left\|\cdot\right\|_q$ is the dual norm of $\left\|\cdot\right\|_p$ such that $\frac{1}{p} + \frac{1}{q} = 1$.

[5]In the paper, we abuse the notation of ρ to present the penalty parameter of an augmented Lagrangian.

Algorithm 4 Computation efficient update of $\mathbf{Z}$.

Require: The number of total iterations K, a positive ρ, and the initial model $\mathbf{Z}_t$.

for $k = 0, 1, 2, ..., K-1$ **do**
  Update $\mathbf{Z}_{k+1}$ by performing Eq. 8.
  Update the m-th column of $\mathbf{W}_{k+1}$ by performing Eq. 9.
  Update $\mathbf{\Omega}_{k+1}$ by performing Eq. 10.
return $\mathbf{Z}_K$.

Algorithm 5 Computation efficient training of personalized models at the server.

Require: The number of total iterations T, and the initial model $\mathbf{x}_1$, and $\mathbf{z}_1^{(n)}$ with $n \in \{1,2,\cdots,N\}$.

Deliver the model $\mathbf{y}_1^{(n)} = \mathbf{M}\mathbf{x}_1 + \mathbf{N}\mathbf{z}_1^{(n)}$ to all client n with $n \in \{1,2,...,N\}$.
for $t = 1, 2, ..., T$ **do**
  for $i = 0, 1, 2, \cdots, I-1$ **do**
    Collect update of local model $\nabla_i = \left[\nabla_t^{(1)}, \nabla_t^{(2)}, ..., \nabla_t^{(N)}\right]$ from all client n with $n \in \{1,2,...,N\}$.
    Update the global model $\mathbf{x}_{t+1}$ by performing:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \eta_i\left(\frac{1}{N}\sum_{n=1}^{N}\mathbf{M}^\top\nabla_i^{(n)}\right).$$

    Deliver the model $\mathbf{y}_{i+1}^{(n)} = \mathbf{M}\mathbf{x}_{i+1} + \mathbf{N}\mathbf{z}_t^{(n)}$ to every client.
  for $j = 0, 1, 2, \cdots, J-1$ **do**
    Collect update of local model $\nabla_j = \left[\nabla_j^{(1)}, \nabla_j^{(2)}, ..., \nabla_j^{(N)}\right]$ from all client n with $n \in \{1,2,...,N\}$.
    Update the personalized model $\mathbf{Z}_{j+1}$ according to Algorithm 4.
    Deliver the parameter $\mathbf{y}_{j+1}^{(n)} = \mathbf{M}\mathbf{x}_I + \mathbf{N}\mathbf{z}_j^{(n)}$ to every client.
return $\mathbf{x}_{T+1}^{(n)} = \mathbf{M}\mathbf{x}_I + \mathbf{N}\mathbf{z}_J^{(n)}$ with $n \in \{1,2,...,N\}$.

Update of Ω. Given $\mathbf{Z}_{k+1}$ and $\mathbf{W}_{k+1}$, $\mathbf{\Omega}_{k+1}$ is obtained by performing:

$$\mathbf{\Omega}_{k+1} = \mathbf{\Omega}_k + \rho\left(\mathbf{Z}_{k+1}\mathbf{Q} - \mathbf{W}_{k+1}\right). \tag{10}$$

Algorithmic details of ADMM are shown in Algorithm 4, and the whole algorithm is illustrated in Algorithm 5. The server conducts update of $\mathbf{Z}$ by ADMM, where all updates of $\mathbf{Z}$, $\mathbf{W}$, and $\mathbf{\Omega}$ can be completed by performing closed-form solutions. Although the update of $\mathbf{Z}$ needs to compute the inverse of matrix, its computational cost can be reduced significantly by using eigenvalue decomposition. Similar to the update of $\mathbf{y}$ in previous section, eigenvectors of $\mathbf{QQ}^\top$ can be reused repeatedly to complete the update of $\mathbf{Z}$. Therefore, it does not lead to high computational cost to find the optimum of $\mathbf{Z}$. Finally, the federated model with personalized and shared components is optimized by performing update of $\mathbf{x}$ and $\mathbf{Z}$ iteratively.

TABLE I
SUMMARY OF EXPERIMENTAL SETTINGS.

Datasets	Models	Tasks	Data types	Metrics
Luna16	D-Net	classification	CT images	test accuracy
BraTS17	U-Net	segmentation	MRI images	test IoU
CHD	TabNet	classification	tabular data	test accuracy
Diabetes	TabNet	classification	tabular data	test accuracy
Covid19	TabNet	classification	tabular data	test accuracy

TABLE II
EVALUATE ACCURACY (%) OF D-NET ON THE DATASET *Luna16*.

Algo.	$\delta = 1$	$\delta = 2$	$\delta = 4$	$\delta = 7$
Ditto	77.42±0.14	73.55±0.64	70.54±2.95	69.25±12.17
FedAMP	72.92±0.29	71.17±0.51	73.74±0.91	80.50±1.09
FedAvg	80.50±0.00	81.12±0.00	77.27±0.00	63.92±8.08
L2GD	73.33±0.14	69.98±0.15	70.20±0.44	81.08±1.04
FedPer	62.33±0.14	66.50±0.15	77.27±0.00	83.92±0.14
FedProx	82.50±1.50	77.89±1.26	72.47±1.54	75.75±1.56
FedRoD	79.50±0.25	79.25±0.53	83.42±1.17	86.00±1.09
FPFC	73.50±0.25	74.91±0.15	78.96±0.15	85.00±0.00
IFCA	80.17±0.29	80.61±0.00	73.74±0.00	63.33±0.52
pFedMe	71.08±0.14	67.43±0.90	71.21±4.59	82.00±0.90
SuPerFed	61.17±0.38	65.31±0.00	76.77±0.00	84.08±0.14
FedRep	77.50±0.50	79.68±0.39	79.97±0.89	83.75±0.00
pFedNet	**86.25±0.00**	**82.74±0.15**	**86.45±0.29**	**86.42±0.14**
rank	**top 1**	**top 1**	**top 1**	**top 1**

VI. EMPIRICAL STUDIES

This section presents performance of the proposed method on model effectiveness, communication efficiency and so on by conducting extensive empirical studies.

A. Experimental Settings

Datasets and tasks. We conduct classification and segmentation tasks on 2 public medical datasets: *Luna16*, *BraTS17*, and 3 private medical datasets collecting from multiple medical centers of hosiptal: *CHD*, *Diabetes*, and *Covid19*. Those datasets own different modalities. Specifically, *Luna16*[6] and *BraTS17*[7] are lung CT and brain tumor MRI images, respectively. *CHD*, *Diabetes*, and *Covid19* are structural medical data[8]. Details of datasets are presented as follows.

- **Luna16**. It is a public dataset to evaluate the algorithmic performance of lung nodule detection. The dataset consists of 888 patients' CT scans, and every scan is sliced into 64 pieces. More than $551,065$ candidates of lung nodules are recognized by tools automatically, while only 1186 true nodules are identified by real doctors. In the experiment, we extract every candidate of lung nodules by using a 32×32 patch.
- **BraTS17**. It is a public dataset, and is usually used to segmentation of glioma sub-regions of brain. The dataset consists of 484 patients' MRI scans, and every scan owns 4 channels. In the experiment, we extract every candidate of brain tumor by using a 64×64 patch.
- **CHD**. The dataset is built from the first medical center of the PLA general hospital of China. It is used to conduct prediction of bleeding risk in elderly patients with coronary heart disease combined with intestinal malignant tumors. The dataset consists of 716 patients' medical records, and every record owns 58 features.
- **Diabetes**. The dataset is built from the first medical center of the PLA general hospital of China, and is used to conduct risk prediction of type 2 diabetes retinopathy. The dataset consists of $31,476$ patients' medical records, and every record owns 63 features.
- **Covid19**. The dataset is built from three medical centers (the first/fifth/sixth medical center) of the PLA general hospital of China, and is used to predict event of Covid-19 infection. The dataset consists of 2402 patients' medical records, and every record owns 77 features.

[6]https://luna16.grand-challenge.org/Data/

[7]https://www.med.upenn.edu/sbia/BraTS17/data.html

[8]It has been approved by the ethics committee of Chinese PLA General Hospital (Grant No. S2022-766-01).

Additionally, we conduct 3 medical analysis tasks, including lung nodule classification, brain tumor segmentation, and clinical risk prediction.

- **Lung nodule classification**. Dense net [52] (D-Net) model is chosen to detect real lung nodules from all candidates. We choose parameters of the fully connecting layer as the personalized component, and others as the shared component.
- **Brain tumor segmentation**. U-Net [53] model is picked to conduct segmentation of brain tumors. Parameters of down-sampling layers are chosen as the shared component, and up-sampling layers' parameters are chosen as the personalized component.
- **Clinical risk prediction**. We use TabNet model [57] to predict whether clinical risks (bleeding, and infection etc.) appears. All features are chosen as the personalized component.

In the experiment, we first fill all missing values by using zeros, and normalize values between -1 and 1. Experimental settings are shown in Table I briefly.

Methods and metrics. The proposed method pFedNet is evaluated by comparing 14 existing methods. Those methods include Ditto [58], FedAMP [20], FedAvg [59], L2GD [28], FedPer [30], FedProx [60], FedRoD [21], APFL [22], FPFC [23], IFCA [24], pFedMe [25], SuPerFed [26], FedRep [27], and FedLC [61]. FedAvg and FedProx are general optimization methods for federated learning, while others are recently proposed personalized federated learning methods. Additionally, the performance of all classification model is measured by test accuracy, and the performance of the segmentation model is measured by Intersection over Union (IoU). These metrics are widely used in previous work [18], [61], [60], [26]. The communication efficiency is measured by the model size.

Federated setting. In the experiment, there are 5 clients and 1 server. The similarity network consists of 5 nodes, and its edges are generated by using KNN with $k = 3$ by default. Heterogeneous data distribution is constructed as follows. Specifically, heterogeneous data federation for classification is built based on the setting of label unbalance, which is measured by $\delta := n_{\text{negative}}/n_{\text{positive}}$. Here, n_{negative} and n_{positive} represent the number of negative and positive

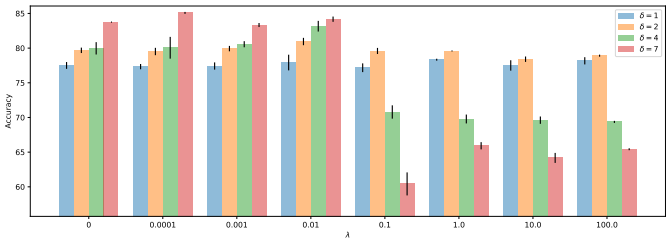
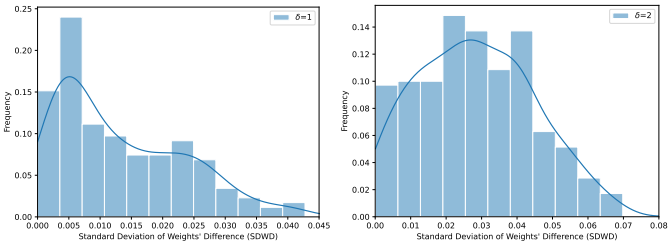
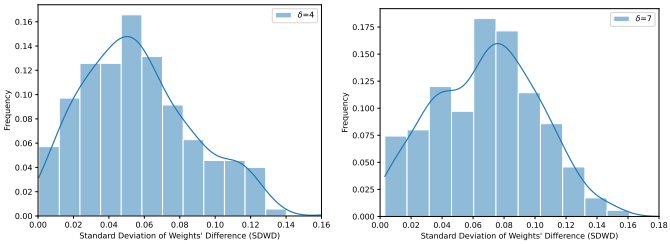


Fig. 6. Test accuracy w.r.t. λ varies significantly on the dataset *Luna16*.

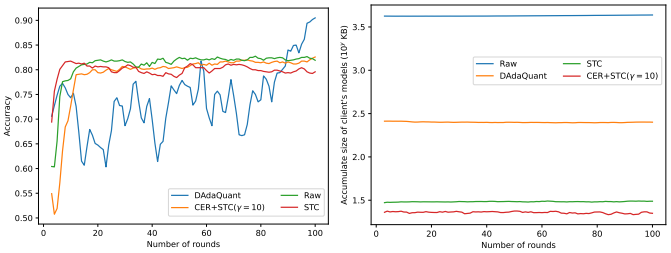


(a) Values of SDWD w.r.t. nodes (b) Values of SDWD w.r.t. nodes

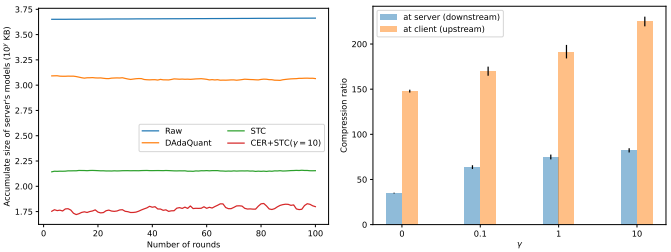


(c) Values of SDWD w.r.t. nodes (d) Values of SDWD w.r.t. nodes

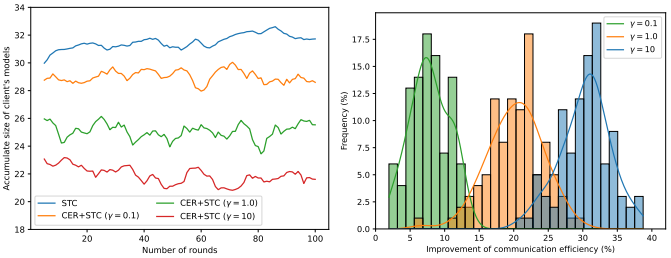
Fig. 7. Values of SDWD has significant variation between nodes. It implies that the personalized component of D-Net has significant difference between nodes.



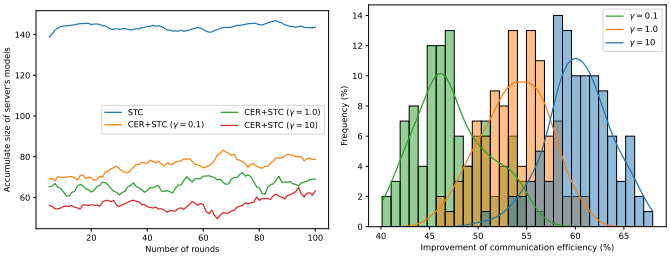
(a) Convergence (b) Accumulated size of clients' model



(c) Accumulated size server's model (d) Model size reduction with CER



(e) Model size reduction at client (f) Statistics of model size reduction at client



(g) Model size reduction at server (h) Statistics of model size reduction at server

Fig. 8. pFedNet significantly reduces model size when performing CER on the dataset *Luna16*.

labels, respectively. In the experiment, we vary δ from $\delta = 1$ to $\delta = 4$ to obtain different settings of data heterogeneity. Similarly, heterogeneous data federation for segmentation is built based on the setting of channel unbalance, which is measured by the id of the missing channel (e.g. *lack #0*, and *lack #1* etc). Additionally, we choose 5-fold cross validation to find an appropriate λ, where 80% of local data is used to train and validate model, and 20% of them is used to test model. In the experiment, we set $\lambda = 0.01$ in the classification task, and choose $\lambda = 1.0$ in the segmentation task by default according to parameter selection. All methods are implemented by using PyTorch, and run on a machine with 24 GB memory, 2TB SSD, and Nvidia Tesla 3090. All numerical values are repeatedly collected by running code with different random seeds, and then report their mean and standard deviation in experimental results.

B. Classification of Lung Nodules

First, we evaluate the model performance of methods, and find pFedNet successfully beat other existing methods. We test personalized model at client, collect all local test accuracy, and then compute the average as the final test accuracy. As illustrated in Table II, pFedNet achieves the best performance, and enjoys more than 3% gains of test accuracy higher than other methods in most cases. Meanwhile, stochastic gradients let the test accuracy vary within $0\% \sim 0.29\%$, which is insignificant to those gains, and does not impair the superiority significantly. Additionally, we vary λ to generate different personalized models. As shown in Figure 6, a small λ tends to yield a more personalized model, which could be adaptive to unbalance data, and obtains higher test accuracy. However, a tiny λ with $\lambda < 10^{-3}$ may falsely view some noise of data as the personalized component, which leads to over-personalized model, and decreases the model performance. It seems that $\lambda = 0.01$ is a good choice since most of unbalanced data achieves best performance. Moreover, we find that test accuracy is mildly sensitive to λ. Comparing with the

different settings of unbalanced data, the sensitivity becomes significant for a large δ. Additionally, we choose Standard Deviation of Weights' Difference (SDWD) with respect to nodes as the metric to evaluate the significance of personalized component. Every parameter of the personalized component has its SDWD. We collect all those values, and show statistics of them in Figure 7. As we can see, since values of SDWD are always larger than 0, it implies that parameters of the personalized component has significant variation between nodes. Besides, such variation becomes noticeable with the increase of δ, and it means that the personalized component is significant for the high heterogeneity of data. It validates the assumption that personalized component is able to capture data's characteristics for every node.

Second, the proposed method, namely CER, successfully improves communication efficiency by reducing model size effectively. Figure 8 shows the superiority of communication efficiency. It is a good complement for existing methods, and can promote their performance effectively. We choose widely used model compression methods including STC [62] and DAdaQuant [63] as the baseline, to show benefits of the proposed method. As shown in Figure 8(a), we observe that CER achieves better accuracy than STC, and more stable convergence than DAdaQuant, respectively. It demonstrates that the superiority on the communication efficiency can be achieved by using CER without significant harm to the test accuracy. Specifically, both Figures 8(b) and 8(c) show the significant effectiveness of CER on compression of models. As illustrated in Figure 8(d). STC without CER achieves up to $146\times$ compression ratio at client, and $34\times$ compression ratio at server, respectively. After equipping with CER ($\gamma = 0.1$), that is CER+STC, successfully achieves up to $172\times$ compression ratio at client and $61\times$ compression ratio at server. The advantage becomes more significant with the increase of γ. As we have claimed, the communication efficiency may be achieved with sacrifice of model performance. As shown in Figure 8(e), CER reduces size of client's model effectively, which becomes more and more significant with the increase of γ. Figure 8(f) shows that CER can improve the communication efficiency at client by reducing $7\% \sim 32\%$ model size more than STC. The benefit becomes more significant when delivering personalized model to every client. Figure 8(g) shows that CER can promote the performance of STC prominently at server, and obtains much more noticeable advantages on the communication efficiency than that at client. Similarly, Figure 8(h) shows that the communication efficiency at server can be improved by reducing $45\% \sim 60\%$ model size more than STC. Although those gains on the communication efficiency have some variations due to stochastic gradients, the effectiveness and superiority of CER is significant. Therefore, those numerical results validate that CER makes a good tradeoff between accuracy and communication efficiency.

Third, we evaluate the robustness of pFedNet by varying similarity network. We build four different networks by running KNN with $1 \leq k \leq 4$. As illustrated in Figure 9, pFedNet has insignificant difference for $k \in \{1, 2, 3\}$, and suffers slight decrease of accuracy for $k = 4$. Since previous numerical results have shown superiority of pFedNet for the setting of $k = 3$, there are reasons to believe that pFedNet enjoys robustness with respect to k.

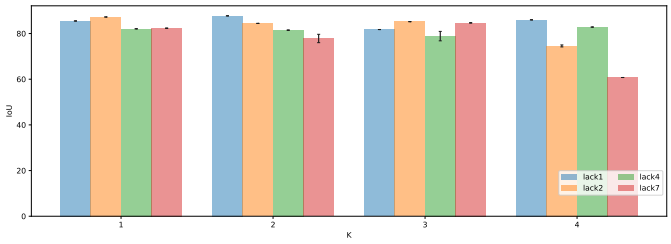


Fig. 9. Test accuracy w.r.t. k varies slightly on the dataset *Luna16*.

TABLE III
EVALUATE *IoU* (%) OF U-NET ON THE DATASET *BraTS17*.

Algo.	lack #0	lack #1	lack #2	lack #3
Ditto	69.80±0.24	69.68±0.25	67.92±0.42	69.07±0.38
FedAMP	67.08±0.12	65.97±0.29	65.42±0.10	67.33±0.61
FedAvg	70.13±0.29	**70.84±0.18**	67.82±0.09	67.29±0.35
L2GD	66.40±0.21	65.77±0.52	65.28±0.25	67.75±0.65
FedPer	61.05±0.90	63.53±0.78	62.22±0.55	65.55±0.17
FedProx	50.65±0.02	52.49±0.00	51.42±0.03	56.77±0.25
FedRoD	68.19±0.31	68.78±0.23	69.04±0.04	69.79±0.23
FPFC	61.98±0.27	61.64±0.36	63.90±0.26	65.81±0.45
IFCA	68.95±0.11	69.72±0.09	67.58±0.14	66.09±0.09
pFedMe	68.78±0.11	67.38±0.45	66.98±0.22	69.14±0.49
SuPerFed	62.67±0.35	63.25±0.32	62.82±0.40	66.78±0.19
FedRep	70.39±0.08	66.80±0.43	70.42±0.25	69.62±0.75
FedLC	68.93±1.41	68.29±0.17	71.27±0.68	69.36±0.15
pFedNet	**70.73±0.26**	70.40±0.04	**71.56±0.21**	**70.63±0.33**
rank	**top 1**	**top 2**	**top 1**	**top 1**

TABLE IV
EVALUATE *label IoU* (%) OF U-NET ON THE DATASET *BraTS17*.

Algo.	lack #0	lack #1	lack #2	lack #3
Ditto	66.25±0.35	65.56±0.10	64.48±0.76	65.73±0.11
FedAMP	63.45±0.30	62.23±0.26	61.71±0.35	63.18±0.24
FedAvg	67.68±0.24	**67.89±0.13**	64.67±0.23	63.72±0.02
L2GD	63.20±0.23	62.27±0.88	61.95±0.59	63.89±0.32
FedPer	56.47±0.89	59.46±0.26	57.30±0.58	58.22±0.26
FedProx	39.16±0.03	42.16±0.14	38.24±0.16	46.07±1.01
FedRoD	64.91±0.32	65.82±0.10	65.46±0.23	66.21±0.10
FPFC	59.12±0.09	59.42±0.08	61.10±0.22	62.63±0.26
IFCA	67.13±0.03	67.19±0.05	64.49±0.33	63.44±0.07
pFedMe	65.16±0.23	64.29±0.22	63.56±0.47	65.05±0.21
SuPerFed	59.38±0.14	59.11±0.66	60.19±0.28	63.07±0.08
FedRep	68.05±0.22	65.10±0.21	68.03±0.21	67.32±0.54
FedLC	64.59±1.27	64.87±0.68	68.16±0.67	65.07±0.36
pFedNet	**71.25±0.19**	67.62±0.26	**69.43±0.10**	**68.61±0.38**
rank	**top 1**	**top 2**	**top 1**	**top 1**

C. Segmentation of Brain Tumor

First, we evaluate *IoU* and *label IoU* for all methods on the dataset *BraTS17* by varying the number of missing channels of MRI images. Here, *IoU* is an evaluation metric used to measure the accuracy of an object detector on a particular dataset. Similarly, *label IoU* measures the accuracy of the foreground of the target object. As shown in Tables III and IV, the proposed method, namely pFedNet, achieves better performance than most of existing methods. Although stochastic

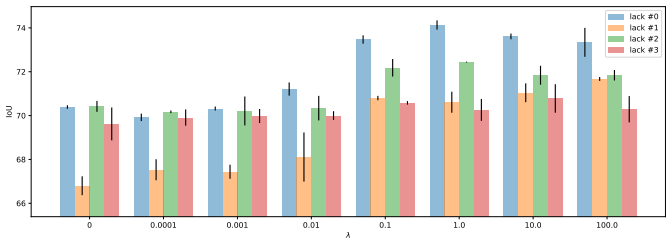
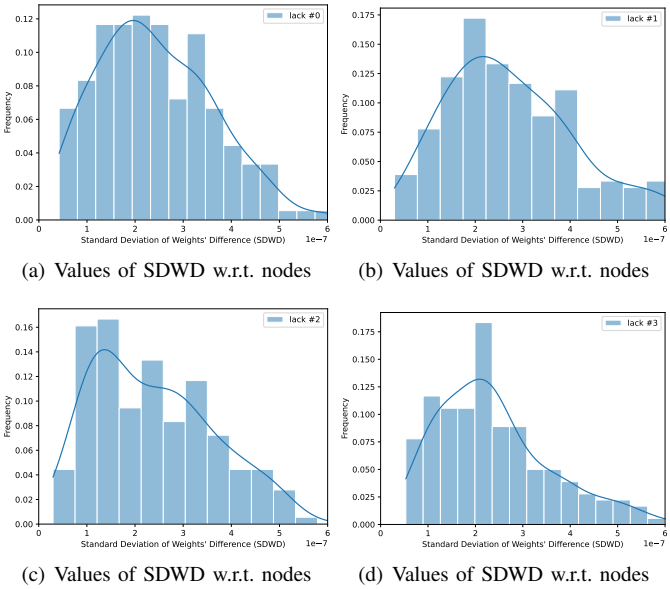


Fig. 10. Test IoU w.r.t. λ varies significantly on the dataset *BraTS17*.



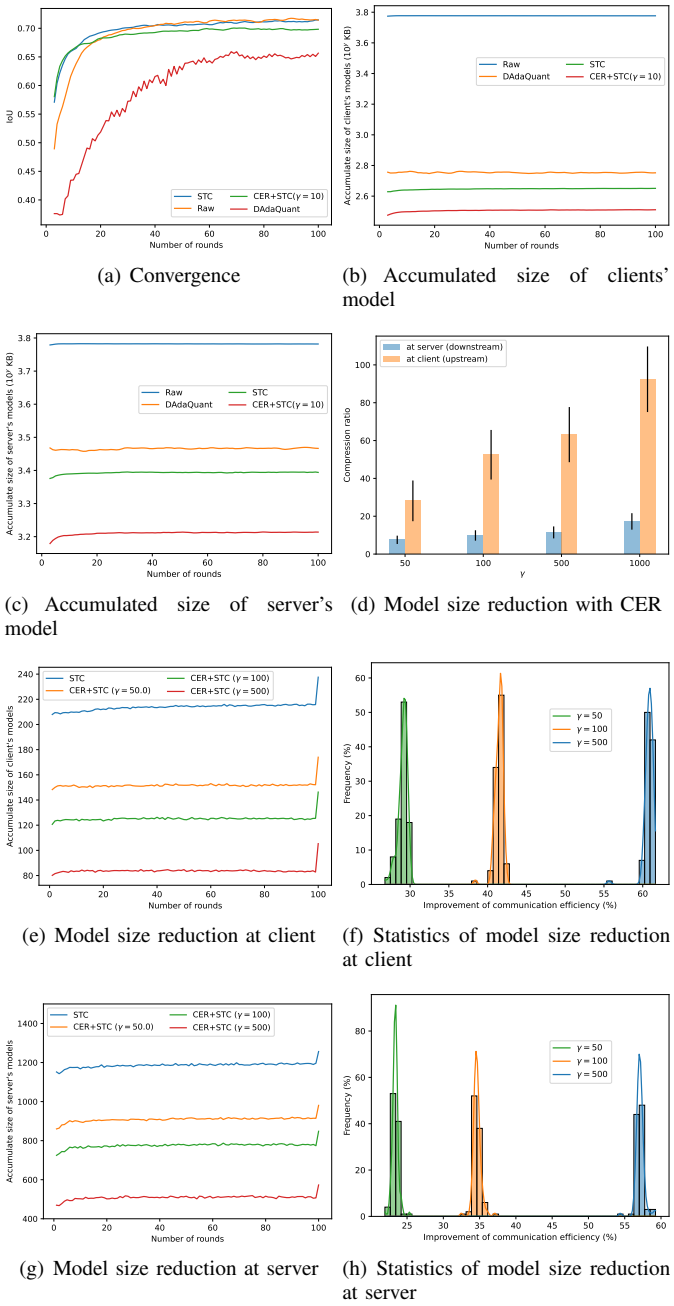
(a) Values of SDWD w.r.t. nodes (b) Values of SDWD w.r.t. nodes

(c) Values of SDWD w.r.t. nodes (d) Values of SDWD w.r.t. nodes

Fig. 11. Values of SDWD has significant variation between nodes. It implies that the personalized component of U-Net has significant difference between nodes.

gradients let the test accuracy vary within $0.04\% \sim 0.38\%$, pFedNet still has noticeable superiority of test accuracy. Specifically, let us focus on the case of 'lack #3' in both Table III and Table IV. Gains of test accuracy achieve 0.8% and 1.3% more than other existing methods respectively. Although pFedNet does not achieve the best performance when missing either #1 or #2 channel, the gap is not significant (less than 0.5%), and still outperforms other methods. Additionally, Figure 10 shows that *IoU* increases significantly with the increase of λ, but may decrease when λ becomes too large. Both *IoU* and *label IoU* are mildly sensitive to λ. We find that $\lambda = 1.0$ seems to be a good choice for different settings of data federation. Moreover, we evaluate the significance of personalized component by using SDWD. As shown in Figure 11, values of SDWD are larger than 0, which implies that the personalized component has significant variation between nodes. It validates that personalized component effectively captures characteristics of local data for every node.

Second, we evaluate the communication efficiency of the proposed method CER. As shown in Figure 12(a), we observe that CER achieves better performance on IoU than DAdaQuant, and shows slightly weak performance than STC, respectively. Figures 12(b) and 12(c) demonstrate that CER owns much more significant advantage on reducing model's



(a) Convergence (b) Accumulated size of clients' model

(c) Accumulated size of server's model (d) Model size reduction with CER

(e) Model size reduction at client (f) Statistics of model size reduction at client

(g) Model size reduction at server (h) Statistics of model size reduction at server

Fig. 12. pFedNet effectively reduces model size with CER on the dataset *BraTS17*.

size than both DAdaQuant and STC, and thus improve communication efficiency effectively. Those observations validate that the superiority on the communication efficiency can be achieved by using CER without significant harm to the performance. As illustrated in Figure 12(d), STC without CER enjoys more than $7\times$ compression ratio of model size at server, and $28\times$ compression ration at client. Although it is effective to compress model, the proposed method CER successfully achieves more than $9\times$ and $52\times$ compression ratio for the server and client when choosing $\gamma = 50$, respectively. Its advantage becomes significant with the increase of γ, and achieves more than $17\times$ and $92\times$ compression ratio for the

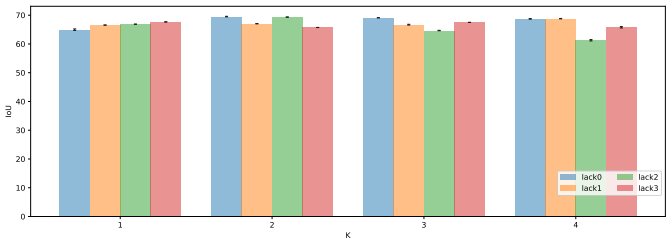


Fig. 13. Test IoU w.r.t. k varies slightly on the dataset *BraTS17*.

server and client. The reason is that CER encourages local update of model to own clustering structure, which is more suitable to conduct compression by using existing methods. We suggest to adopt the dynamic strategy to choosing γ during model learning to obtain more gains of communication efficiency without much sacrifice of model performance. Figure 12(e) shows that CER can be used together with STC, and achieves much more significant compression. The superiority becomes significant with increase of γ. According to Figure 12(f), we observe that CER gains more than $28\% \sim 60\%$ improvement of communication efficiency at client. Similarly, we find that more than $23\% \sim 57\%$ improvement of communication efficiency at server according to Figures 12(g) and 12(h). Although those gains on the communication efficiency have some variations due to stochastic gradients, CER still shows significant superiority of communication efficiency. It validates that CER can successfully find a good tradeoff between model performance and communication efficiency once more.

Third, we vary k with $1 \leq k \leq 4$ to test the robustness of pFedNet. Recall that the similarity network is built by running KNN with a given k, and we build four different similarity networks. As shown in Figure 13, pFedNet has unnoticeable difference for different settings of k. It demonstrates that pFedNet works well for different similarity networks, and enjoys robustness with respect to k. Finally, Figure 14 illustrates the true region of target object (the red line), and some examples of the segmentation region (the blue line) yielded by pFedNet and other methods for the setting of 'lack #0'. As we can see, pFedNet captures details of interested region more accurately than others.

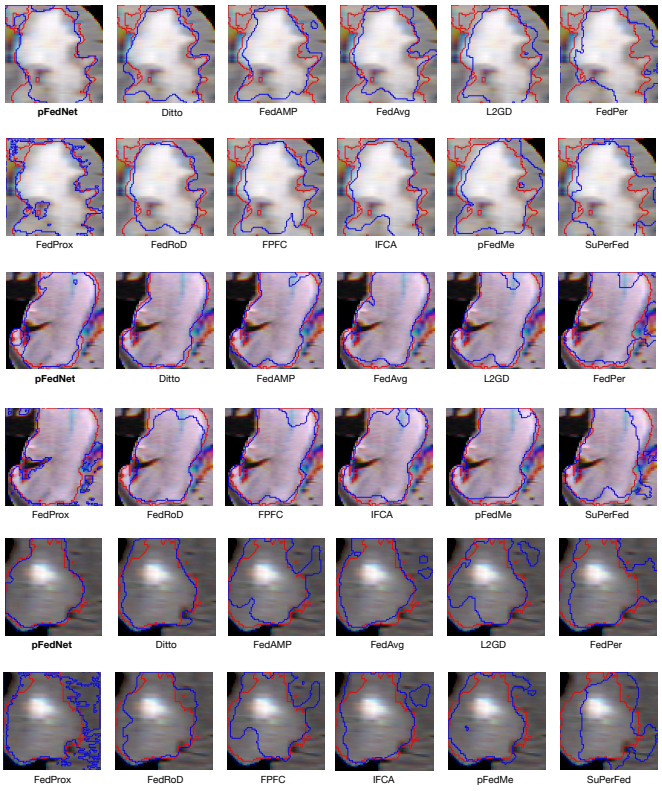


Fig. 14. Segmentation of brain tumors are illustrated as examples by using pFedNet and other existing methods. Red line represents the true region. Blue line represents the segmentation region.

TABLE V
EVALUATE ACCURACY (%) OF TABNET ON THE DATASET *CHD*.

Algo.	δ=1	δ=2	δ=3	δ=4
Ditto	58.33±10.41	50.88±6.08	61.11±0.00	48.15±6.42
FedAMP	43.33±5.77	75.44±3.04	68.52±3.21	55.56±0.00
FedAvg	23.33±2.89	47.37±5.26	44.44±5.56	33.33±0.00
L2GD	60.00±5.00	43.86±6.08	57.41±3.21	51.85±3.21
FedPer	30.00±0.00	26.32±0.00	50.00±0.00	72.22±0.00
FedProx	38.33±5.77	52.63±5.26	24.07±3.21	48.15±3.21
FedRoD	61.67±5.77	52.63±5.26	38.89±5.56	55.56±0.00
FPFC	46.67±2.89	50.88±3.04	55.56±0.00	72.22±0.00
IFCA	**73.33±2.89**	42.11±0.00	48.15±3.21	55.56±0.00
pFedME	40.00±0.00	71.93±12.15	59.26±3.21	64.81±3.21
SuPerFed	40.00±0.00	57.89±0.00	50.00±0.00	68.52±3.21
FedRep	41.67±2.89	49.12±3.04	55.56±0.00	57.41±8.49
pFedNet	**73.33±2.89**	**82.46±3.04**	**72.22±0.00**	**74.07±3.21**
rank	**top 1**	**top 1**	**top 1**	**top 1**

D. Prediction of Clinical Risk

We evaluate pFedNet by using TabNet on three structural medical datasets, which are collected from real scenarios of clinical diagnosis and treatment. As illustrated in Tables V-VII, pFedNet outperforms other methods on the test accuracy in most settings of δ. It shows that the proposed model pFedNet works well for tabular data, and validates the superiority on the model performance again. Note that this advantage is obtained with $\lambda = 0.01$, which is usually a good choice from observation of previous experiments. The promotion of communication efficiency leaded by CER is also validated in the experiment, which is not presented here due to the length limit.

VII. CONCLUSION

We propose a new formulation of personalized federated learning, which has good adaption to heterogenous medical data, and achieves better performance than existing methods. To improve the communication efficiency, we further develop a communicate efficient regularizer , which can decrease workload of communication effectively. Additionally, we propose a new optimization framework to update personalized models, which reduces computation cost significantly. Extensive empirical studies have been conducted to verify the effectiveness of the proposed method. In the future, we explore and analyze the dynamics of medical data, and try to develop the adaptive version of the proposed model to capture such dynamics.

TABLE VI
EVALUATE ACCURACY (%) OF TABNET ON THE DATASET *Diabetes*.

Algo.	δ=1	δ=2	δ=3	δ=4
Ditto	68.02±1.58	72.70±3.98	76.50±0.31	70.04±3.58
FedAMP	73.74±0.74	**77.22±1.19**	77.28±0.65	**79.22±0.56**
FedAvg	73.58±0.43	69.91±5.09	72.10±0.54	69.38±1.90
L2GD	71.45±2.53	75.66±1.16	77.90±0.86	78.40±1.13
FedPer	65.73±5.58	70.36±0.91	74.16±0.56	78.89±0.25
FedProx	57.91±0.88	54.43±1.94	58.35±0.58	52.96±0.93
FedRoD	71.53±2.21	72.70±2.65	72.76±4.39	79.38±1.29
FPFC	62.70±1.00	66.22±0.89	73.09±1.86	78.19±0.47
IFCA	70.55±0.56	68.35±1.07	71.03±7.00	74.49±2.49
pFedME	64.21±1.07	70.61±0.14	75.60±0.78	78.31±1.48
SuPerFed	51.33±1.67	56.98±8.16	68.60±3.53	61.93±1.39
FedRep	69.41±1.02	70.24±1.21	74.65±0.38	77.00±0.50
pFedNet	**74.68±1.97**	76.03±1.20	**78.02±0.81**	78.60±0.47
rank	**top 1**	**top 2**	**top 1**	**top 3**

TABLE VII
EVALUATE ACCURACY (%) OF TABNET ON THE DATASET *Covid19*.

Algo.	δ=1	δ=2	δ=3	δ=4
Ditto	71.67±2.89	72.22±7.27	49.59±1.41	70.00±0.00
FedAMP	71.67±1.44	74.60±1.37	69.92±2.82	80.83±1.44
FedAvg	79.17±3.82	68.25±7.27	58.54±2.44	60.83±1.44
L2GD	73.33±3.82	63.49±7.65	64.23±3.73	76.67±3.82
FedPer	52.50±2.50	41.27±1.37	67.48±1.41	76.67±1.44
FedProx	79.17±1.44	65.08±4.96	54.47±1.41	55.83±5.20
FedRoD	74.17±5.20	**80.16±2.75**	62.60±1.41	70.00±0.00
FPFC	70.00±2.50	70.63±3.64	69.11±1.41	81.67±1.44
IFCA	81.67±1.44	69.05±2.38	68.29±0.00	53.33±12.58
pFedME	80.00±0.00	69.84±1.37	69.92±1.41	72.50±5.00
SuPerFed	70.00±2.50	53.97±1.37	67.48±1.41	71.67±1.44
FedRep	50.83±6.29	57.14±2.38	62.60±1.41	68.33±1.44
pFedNet	**82.50±0.00**	80.16±3.64	**72.36±1.41**	**86.67±3.82**
rank	**top 1**	**top 2**	**top 1**	**top 1**

ACKNOWLEDGMENT

All studies of this work have been approved by the ethics committee of Chinese PLA General Hospital (Grant No. S2022-766-01). This work was supported by the National Natural Science Foundation of China (Grant No. 62302522), the funding Grant No. 22-TQ23-14-ZD-01-001, and the funding Grant No. 145BQ090003000X03. Zongren Li, Qin Zhong, and Hebin Che provide help on collection of medical datasets. Thanks a lot for their kind heart and help.

REFERENCES

[1] D. Placido, B. Yuan, J. X. Hjaltelin, C. Zheng, A. D. Haue, P. J. Chmura, C. Yuan, J. Kim, R. Umeton, G. Antell, A. Chowdhury, A. Franz, L. Brais, E. Andrews, D. S. Marks, A. Regev, S. Ayandeh, M. T. Brophy, N. V. Do, P. Kraft, B. M. Wolpin, M. H. Rosenthal, N. R. Fillmore, S. Brunak, and C. Sander, "A deep learning algorithm to predict risk of pancreatic cancer from disease trajectories," *Nature medicine*, May 2023.

[2] A. H. Thieme, Y. Zheng, G. Machiraju, C. Sadee, M. Mittermaier, M. Gertler, J. L. Salinas, K. Srinivasan, P. Gyawali, F. Carrillo-Perez, A. Capodici, M. Uhlig, D. Habenicht, A. Löser, M. Kohler, M. Schuessler, D. Kaul, J. Gollrad, J. Ma, C. Lippert, K. Billick, I. Bogoch, T. Hernandez-Boussard, P. Geldsetzer, and O. Gevaert, "A deep-learning algorithm to classify skin lesions from mpox virus infection," *Nature Medicine*, vol. 29, no. 3, p. 738—747, March 2023.

[3] W. Chen, R. Li, Q. Yu, A. Xu, Y. Feng, R. Wang, L. Zhao, Z. Lin, Y. Yang, D. Lin, X. Wu, J. Chen, Z. Liu, Y. Wu, K. Dang, K. Qiu, Z. Wang, Z. Zhou, D. Liu, Q. Wu, M. Li, Y. Xiang, X. Li, Z. Lin, D. Zeng, Y. Huang, S. Mo, X. Huang, S. Sun, J. Hu, J. Zhao, M. Wei, S. Hu, L. Chen, B. Dai, H. Yang, D. Huang, X. Lin, L. Liang, X. Ding, Y. Yang, P. Wu, F. Zheng, N. Stanojcic, J.-P. O. Li, C. Y. Cheung, E. Long, C. Chen, Y. Zhu, P. Yu-Wai-Man, R. Wang, W.-S. Zheng, X. Ding, and H. Lin, "Early detection of visual impairment in young children using a smartphone-based deep learning system," *Nature Medicine*, vol. 29, no. 2, p. 493—503, February 2023.

[4] J. Lipkova, T. Y. Chen, M. Y. Lu, R. J. Chen, M. Shady, M. Williams, J. Wang, Z. Noor, R. N. Mitchell, M. Turan, G. Coskun, F. Yilmaz, D. Demir, D. Nart, K. Basak, N. Turhan, S. Ozkara, Y. Banz, K. E. Odening, and F. Mahmood, "Deep learning-enabled assessment of cardiac allograft rejection from endomyocardial biopsies," *Nature Medicine*, vol. 28, no. 3, p. 575—582, March 2022.

[5] M. Gehrung, M. Crispin-Ortuzar, A. G. Berman, M. O'Donovan, R. C. Fitzgerald, and F. Markowetz, "Triage-driven diagnosis of barrett's esophagus for early detection of esophageal adenocarcinoma using deep learning," *Nature Medicine*, vol. 27, no. 5, pp. 833–841, 2021.

[6] A. Brauneck, L. Schmalhorst, M. M. Kazemi Majdabadi, M. Bakhtiari, U. Völker, C. C. Saak, J. Baumbach, L. Baumbach, and G. Buchholtz, "Federated machine learning in data-protection-compliant research," *Nature Machine Intelligence*, vol. 5, no. 1, pp. 2–4, 2023.

[7] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz *et al.*, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

[8] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 118, no. 17, 2021.

[9] C. Wu, F. Wu, L. Lyu, T. Qi, Y. Huang, and X. Xie, "A federated graph neural network framework for privacy-preserving personalization," *Nature Communications*, vol. 13, no. 1, p. 3091, 2022.

[10] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature Communications*, vol. 13, no. 1, p. 2032, 2022.

[11] D. Froelicher, J. R. Troncoso-Pastoriza, J. L. Raisaro, M. A. Cuendet, J. S. Sousa, H. Cho, B. Berger, J. Fellay, and J.-P. Hubaux, "Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption," *Nature Communications*, vol. 12, no. 1, p. 5910, 2021.

[12] J. Terrail, A. Leopold, C. Joly, C. Béguier, M. Andreux, C. Maussion, B. Schmauch, E. Tramel, E. Bendjebbar, M. Zaslavskiy, G. Wainrib, M. Milder, J. Gervasoni, J. Guerin, T. Durand, A. Livartowski, K. Moutet, C. Gautier, I. Djafar, and P.-E. Heudel, "Federated learning for predicting histological response to neoadjuvant chemotherapy in triple-negative breast cancer," *Nature Medicine*, vol. 29, pp. 1–12, 01 2023.

[13] S. Pati, U. Baid, B. Edwards, M. Sheller, S.-H. Wang, G. A. Reina, P. Foley, A. Gruzdev, D. Karkada, C. Davatzikos *et al.*, "Federated learning enables big data for rare cancer boundary detection," *Nature Communications*, vol. 13, no. 1, p. 7346, 2022.

[14] C. I. Bercea, B. Wiestler, D. Rueckert, and S. Albarqouni, "Federated disentangled representation learning for unsupervised brain anomaly detection," *Nature Machine Intelligence*, vol. 4, no. 8, pp. 685–695, 2022.

[15] I. Dayan, H. Roth, A. Zhong, A. Harouni, A. Gentili, A. Abidin, A. Liu, A. Costa, B. Wood, C.-S. Tsai, C.-H. Wang, C.-N. Hsu, C. Lee, P. Ruan, D. Xu, D. Wu, E. Huang, F. Kitamura, G. Lacey, and Q. Li, "Federated learning for predicting clinical outcomes in patients with covid-19," *Nature Medicine*, vol. 27, pp. 1–9, 10 2021.

[16] X. Bai, H. Wang, M. Liya, Y. Xu, J. Gan, Z. Fan, F. Yang, K. Ma, J. Yang, S. Bai, C. Shu, X. Zou, R. Huang, C. Zhang, X. Liu, D. Tu, C. Xu, W. Zhang, X. Wang, and T. Xia, "Advancing covid-19 diagnosis with privacy-preserving collaboration in artificial intelligence," *Nature Machine Intelligence*, vol. 3, 12 2021.

[17] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[18] Z. Chen, C. Yang, M. Zhu, Z. Peng, and Y. Yuan, "Personalized retrogress-resilient federated learning toward imbalanced medical data," *IEEE Transactions on Medical Imaging (TMI)*, vol. 41, no. 12, pp. 3663–3674, 2022.

[19] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2022.

[20] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 7865–7873.

[21] H.-Y. Chen and W.-L. Chao, "On bridging generic and personalized federated learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

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

[23] X. Yu, Z. Liu, Y. Sun, and W. Wang, "Clustered federated learning based on nonconvex pairwise fusion," *arXiv preprint arXiv:2211.04218*, 2022.

[24] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *IEEE Transactions on Information Theory (TOIT)*, vol. 68, no. 12, pp. 8076–8091, 2022.

[25] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," in *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Red Hook, NY, USA, 2020.

[26] S.-J. Hahn, M. Jeong, and J. Lee, "Connecting low-loss subspace for personalized federated learning," in *Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2022, p. 505–515.

[27] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021, pp. 2089–2099.

[28] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," *arXiv preprint arXiv:2002.05516*, 2020.

[29] I. S. Chan and G. S. Ginsburg, "Personalized medicine: progress and promise," *Annual Review of Genomics and Human Genetics*, vol. 12, pp. 217–244, 2011.

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

[31] D. Bui, K. Malik, J. Goetz, H. Liu, S. Moon, A. Kumar, and K. G. Shin, "Federated user representation learning," *arXiv preprint arXiv:1909.12535*, 2019.

[32] 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.

[33] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[34] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021, pp. 12 878–12 889.

[35] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 2351–2363, 2020.

[36] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large cnns at the edge," *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 14 068–14 080, 2020.

[37] I. Bistritz, A. Mann, and N. Bambos, "Distributed distillation for on-device learning," *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 22 593–22 604, 2020.

[38] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Proceedings of the Advances in neural information processing systems (NeurIPS)*, vol. 30, 2017.

[39] L. Corinzia, A. Beuret, and J. M. Buhmann, "Variational federated multi-task learning," *arXiv preprint arXiv:1906.06268*, 2019.

[40] N. Shoham, T. Avidor, A. Keren, N. Israel, D. Benditkis, L. Mor-Yosef, and I. Zeitak, "Overcoming forgetting in federated learning on non-iid data," *arXiv preprint arXiv:1910.07796*, 2019.

[41] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," *arXiv preprint arXiv:2010.01264*, 2020.

[42] 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 (TNNLS)*, vol. 32, no. 8, pp. 3710–3722, 2020.

[43] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–9.

[44] M. Duan, D. Liu, X. Ji, R. Liu, L. Liang, X. Chen, and Y. Tan, "Fedgroup: Efficient federated learning via decomposed similarity-based clustering," in *Proceedings of the IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking*, 2021, pp. 228–237.

[45] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 19 586–19 597, 2020.

[46] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *Journal of Biomedical Informatics*, vol. 99, p. 103291, 2019.

[47] B. Pfitzner, N. Steckhan, and B. Arnrich, "Federated learning in a medical context: A systematic literature review," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 2, pp. 1–31, 2021.

[48] G. Kaissis, A. Ziller, J. Passerat-Palmbach, T. Ryffel, D. Usynin, A. Trask, I. Lima Jr, J. Mancuso, F. Jungmann, M.-M. Steinborn *et al.*, "End-to-end privacy preserving deep learning on multi-institutional medical imaging," *Nature Machine Intelligence*, vol. 3, no. 6, pp. 473–484, 2021.

[49] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, 2020.

[50] W. Zhu and J. Luo, "Federated medical image analysis with virtual sample synthesis," in *Proceedings of the Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Springer, 2022, pp. 728–738.

[51] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, USA, 2014.

[52] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[53] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[54] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[55] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, p. 1–122, jan 2011.

[56] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, p. 127–239, jan 2014.

[57] S. O. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, May 2021, pp. 6679–6687.

[58] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International Conference on Machine Learning (ICML)*, 2020.

[59] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[60] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[61] J. Wang, Y. Jin, and L. Wang, "Personalizing federated medical image segmentation via local calibration," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 456–472.

[62] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, vol. 31, no. 9, pp. 3400–3413, 2019.

[63] R. Hönig, Y. Zhao, and R. Mullins, "Dadaquant: Doubly-adaptive quantization for communication-efficient federated learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022, pp. 8852–8866.

Yawei Zhao is now working at Medical Big Data Research Center of Chinese PLA General Hospital & National Engineering Research Center for the Application Technology of Medical Big Data, Beijing, 100853, China. He received his Ph.D, B.E., and M.S. degree in Computer Science from the National University of Defense Technology, China, in 2013, 2015, and 2020, respectively. His research interests include federated learning, and medical artificial intelligence. Dr. Zhao has published 10+ peer-reviewed papers in highly regarded journals and conferences such as IEEE T-KDE, IEEE T-NNLS, AAAI, etc.

Qinghe Liu is now pursuing his M.S. degree at Chinese PLA General Hospital & National Engineering Research Center for the Application Technology of Medical Big Data, Beijing, 100853, China. His research interests include federated learning and medical image analysis.

Pan Liu received his PhD. degrees in Department of Mathematics, Non-linear Analysis Center, of Carnegie Mellon University. He then was working as a research associate in the Department of Pure Mathematics and Mathematical Statistics at the University of Cambridge. He is now a postdoc fellow in the Medical Big Data Research Center, Chinese PLA General Hospital. His interests include machine learning and computer vision, with focus on deep learning and medical image processing.

Xinwang Liu received his Ph.D. degree from the National University of Defense Technology (NUDT), China. He is now a full professor at the College of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 100+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, ICML, NeurIPS, ICCV, CVPR, AAAI, IJCAI, etc. He is a senior member of IEEE. He serves as the associated editor of the Information Fusion Journal, IEEE T-NNLS Journal, and IEEE T-CYB Journal. More information can be found at https://xinwangliu.github.io.

Kunlun He received his M.D. degree from The 3rd Military Medical University, Chongqing, China in 1988, and PhD degree in Cardiology from Chinese PLA Medical school, Beijing, China in 1999. He worked as a postdoctoral research fellow at Division of circulatory physiology of Columbia University from 1999 to 2003. He is director and professor of Medical Big Data Research Center, Chinese PLA General Hospital, Beijing, China. His research interests include big data and artificial intelligence of cardiovascular disease.