Dual-Objective Personalized Federated Service System with Partially-labeled Data over Wireless Networks

Cheng-Wei Ching, Jia-Ming Chang, Jian-Jhih Kuo, *Member, IEEE*, and Chih-Yu Wang, *Senior Member, IEEE*

Abstract—Federated learning (FL) emerges to mitigate the privacy concerns in machine learning-based services and applications, and personalized federated learning (PFL) evolves to alleviate the issue of data heterogeneity. However, FL and PFL usually rest on two assumptions: the users' data is well-labeled, or the personalized goals align with sufficient local data. Unfortunately, the two assumptions may not hold in most cases, where data labeling is costly, or most users have no sufficient local data to satisfy their personalized needs. To this end, we first formulate the problem, DoLP, that studies the issue of insufficient and partially-labeled data on FL-based services. DoLP aims to maximize two service objectives: 1) personalized classification objective and 2) the personalized labeling objective for each user within the constraint of training time over wireless networks. Then, we propose a PFL-based service system DoFed-SPP to solve DoLP. The DoFed-SPP's novelty is two-fold. First, we devise an inference-based first-order approximation metric, similarity ratio, to identify the similarity between users' local data. Second, we design an approximation algorithm to determine the appropriate size and set of users for uploading in each round. Extensive experiments show DoFed-SPP outperforms the state-of-the-art in final accuracy and time-to-accuracy performance on CIFAR10/100 and DBPedia.

Index Terms—Federated Learning Service System, Personalized Federated Learning Service System, Personalized Services, Data Labeling Services, Data Classification Services, Heterogeneous User Data, Partially-labeled Data, Similarity Ratios, Approximation Algorithm.

1 Introduction

Nowadays, the success of machine learning-based services heavily relies on the massive amount of data. With the growth of smart devices, a significant amount of data, such as photos, voices, and positions, is generated in a timely manner and can help model training. However, due to privacy concerns, we cannot collect data from devices in many circumstances. Thus, *federated learning* (FL) is proposed to address the privacy concerns [1]. With FL, a group of participants train their local models with their own data and send their local models to the *parameter server* (PS), and PS aggregates these local models to update the global model. Not only does FL allow training a model jointly with less privacy concerns, but FL is adaptive to large-scale systems. FL has achieved great success in several applications and services, such as voice recognition [2], [3], natural language processing [4], [5], recommendation systems [6], [7], word candidate prediction [6], [8], and healthcare [9], [10] .

- *C.-W. Ching is with the Department of Computer Science and Engineering, UCSC, USA, and the Research Center for Information Technology Innovation, Academia Sinica, Taiwan. E-mail: cching1@ucsc.edu.*
- *J.-M. Chang is with the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan. E-mail: g609410078@alum.ccu.edu.tw*
- *J.-J. Kuo is with the Department of Computer Science and Information Engineering and the Advanced Institute of Manufacturing with High-tech Innovations, National Chung Cheng University, Taiwan. E-mail: lajacky@cs.ccu.edu.tw.*
- *C.-Y. Wang is with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan. E-mail: cywang@citi.sinica.edu.tw.*

The conventional FL aims to *minimize the loss of the aggregate of local models*. In particular, in heterogeneous settings where the underlying data distribution of the user data is usually divergent, the final global model obtained by minimizing the average loss (see Section 2 for more details) might perform rather poorly once applied to the local data of each user [11]–[13]. To this end, *personalized federated learning* (PFL) is proposed to mitigate the negative impact of heterogeneous settings [11], [13], [14]. Compared to FL, PFL aims at minimizing the loss of each local model with respect to their local data in a federated manner in hope of improving personalized service experiences.

However, the existing PFL and FL work usually rests on the assumption that *the users' data is all well-labeled*. They focus more on the supervised learning, where the local data is usually well and correctly labeled. Nonetheless, such an assumption is rather unrealistic in numerous applications and services since labeling is considered time-consuming and costly [15]–[18]. Practically, users are more likely to have a certain amount of labeled data that they can easily obtain and another amount of partially-labeled data that they have difficulty in fully labeling [13], [19], [20]. Moreover, *the PFL further assumes the personalized goals align with the sufficient local data*. The existing PFL work usually assumes that the users' personalized goals are to find a model best fit for all of their local data, and the users can enhance the generalized performance of personalized goals by participating PFL. Unfortunately, users in practice are likely to have only a small amount of data benefit to their personalized goals [14], [21], [22]. Under the circumstances, the final person-

alized models that the users obtain from participating the existing PFL systems may not reach their expectations. For practical personalized model training, a better collaboration approach is required to relax these two assumptions.

To study the impact of insufficient and partially-labeled data on personalized model training, we firstly propose the **D**ual-**o**bjective **L**earning **P**roblem with partially-labeled data within the constraint of training service time (DoLP) as the main goal of this paper. Specifically, we consider the service environment over wireless networks, where there are a set of devices (i.e., the users) and one *base station* (BS) that coordinates the wireless transmission of the users. The users each have a training dataset and a target dataset, where the data distributions of the two datasets are *heterogeneous*. The data points in the training dataset are well labeled, whereas those in the target dataset are partially-labeled. The users each have two service objectives. One is the *personalized classification objective* that asks for a model best fit for their labeled data points in the target dataset. On the contrary, the other is *personalized labeling objective* which aims to find another model that labels the unlabeled data points in the target dataset as correctly as possible. Meanwhile, a user-defined training service time composed of on-device computation and wireless transmission necessary for training is given such that the necessary training time does not exceed the given threshold.

Addressing DoLP gives rise to three challenges as follows. The first challenge is *heterogeneous data distributions between the training dataset and the target dataset*. Since the data distributions of the two datasets may be significantly heterogeneous, FL or PFL is expected to perform poorly for them, as models trained using FL or PFL are for their labeled training datasets only, and there is no guarantee for these models to perform well on their target dataset as well as the training dataset in either classification or labeling services. Then, the second challenge is *training on the small and partially-labeled target dataset*. Conventionally, this can be considered as *semi-supervised learning* (SSL), which refers to a learning problem with partially-labeled data, where the ratio of unlabeled data is usually much larger than that of the labeled data (e.g., 1 to 10), and the objective of SSL is to label the unlabeled data as correctly as possible. As we have assumed that the dataset owned by each user is insufficient to train the models, conventional SSL cannot be applied [23]. *Federated semi-supervised learning* (FSSL) is proposed recently to improve the generalized performance of SSL by aggregating local models from different users [20]. However, FSSL aims at generalized performance over all users [20] rather than personalized performance, which is incompatible with the personalized labeling objective. The last challenge is *unrepresentative training and testing performance*. When training a machine learning model for a specific application or service, the data is usually split into two disjoint sets. One is the training set that accounts for most of the data. The other is the testing set. The training set is used for training the model from scratch, whereas the testing set serves to measure the performance of the model during training. However, the users are short of a large amount of labeled target data, implying that they can neither perform local training with respect to the personalized classification objectives nor measure how well a model trained via FL or PFL performs in term of the personalized classification objective (see Section 6 for more details).

To this end, we propose **D**ual-**o**bjective **Fed**erated Learning System for the **S**ervices of **P**ersonalization and **P**artially-labeled Data (DoFed-SPP) to address DoLP and the three challenges. The key innovation is that DoFed-SPP proposes a complementary training scheme, in which users each train a reciprocal model fit for their training data and use the first-order approximation-based *similarity ratios* (SRs) to download the relevant reciprocal models of other users. Then, the users each exploit the downloaded reciprocal models to label the unlabeled data points in the target dataset *in an ensemble-based manner* for the personalized labeling objective. Lastly, we adopt the teacher-student architecture to train the target model for the personalized classification objective. Specifically, the users each establish a pseudo-labeled dataset composed of the original labeled data points and the pseudo-labeled data points in the target datasets (i.e., labeled by the downloaded reciprocal models), and execute local training using an initialized model (i.e., the student) with the pseudo-labeled dataset (i.e., the teacher) for the personalized classification objective. For the constraint of training time, we first break it down into the computation time and the communication time, where the former is estimated by dividing required CPU cycles by computational capacity of devices, and the latter is estimated by dividing the model size by Shannon capacity [24] given the channel quality in wireless networks. Based on the estimations, we formulate an optimization problem to determine the number and the set of reciprocal models downloaded, and then propose a matching algorithm which achieves $(1 - \frac{1}{e})$ approximation ratio in a model accuracy-dependent indicator (refer to Section 4.4 for more details).

On the implementational side, we consider two performance metrics. One is the final accuracy, and the other is the time-to-accuracy performance. The experimental results show that DoFed-SPP makes significant improvement on both performance metrics using three benchmarks, CIFAR10, CIFAR100, and DBPedia. Moreover, we conduct extensive experiments on the parameters of DoFed-SPP to show DoFed-SPP is robust to different settings.

The contributions of this paper are summarized as follows.

- We make the first attempt to study the impact of insufficient and partially-labeled data to PFL, and thereby formulate the problem DoLP that aims to maximize two personalized service objectives: classification and labeling accuracy, within the constraint of training time.
- We propose a novel PFL-based service system DoFed-SPP to solve DoLP. DoFed-SPP adopts an inference-based first order approximation metric, similarity ratios, to identify the similarity between the local data of devices. Meanwhile, DoFed-SPP uses an approximation algorithm with a theoretical proof to select a near-optimal set of devices in each communication round to upload local models in order to maximize the two personalized objectives within the constraint of training time.
- Extensive experiments show that DoFed-SPP outperforms two state-of-the-art methods, FedFomo [14]

and FedBE [5], and the conventional one, FedAvg [1], in terms of final accuracy and time-to-accuracy performance, using three benchmarks, CIFAR10, CIFAR100, and DBPedia.

The rest of this paper is organized as follows. Section 2 introduces the preliminaries. Section 3 first details the system model, and then defines DoLP. The proposed system DoFed-SPP is presented in Section 4. Section 5 summarizes the workflow of DoFed-SPP. Section 6 shows the experimental results. The related work is reviewed in Section 7. Section 8 concludes this paper.

2 Preliminaries

2.1 Federated Learning (FL)

In conventional FL, there are N users and a PS that aim to solve the following problem:

$$\min_{\Theta\in\mathbb{R}^d} f(\Theta) \coloneqq \frac{1}{N}\sum_{n=1}^{N} f_n(\Theta), \tag{1}$$

where Θ is the global model. The function $f_n(\Theta)$ for each $n \in [N]$ denotes the expected loss over the data distribution of user n:

$$f_n(\Theta) \coloneqq \mathbb{E}_{\xi_n}[F_n(\Theta;\xi_n)], \tag{2}$$

where ξ_n is a random data sample drawn according to the distribution of user n and $F_n(\Theta;\xi_n)$ is a loss function corresponding to this sample and Θ.

In practice, the user $n \in [N]$ has a set of data $\mathcal{D}_n$. In communication round t, a set of users $[K] \subset [N]$ are sampled randomly, each of which downloads the global model Θ^t (i.e., the global model in communication round t) from PS. Then, each sampled user $k \in [K]$ performs the following local training with its local data $\mathcal{D}_k$,

$$\theta_k^{t+1} = \Theta^t - \gamma\nabla_{\Theta^t} f_k(\mathcal{D}_k;\Theta^t), \tag{3}$$

where θ_k^{t+1} denotes the local model of user k in communication round t and $\gamma \in (0,1]$ denotes the learning rate. Then, users upload local models to PS. Upon collecting K local models, PS performs aggregate to attain updated global model Θ^{t+1} as follows:

$$\Theta^{t+1} = \sum_{k\in[K]} w_k\theta_k^{t+1}, \tag{4}$$

where $w_k \coloneqq \frac{|\mathcal{D}_k|}{\sum_{j\in K}|\mathcal{D}_j|}$. Remark that users can upload the gradients $\nabla_{\Theta^t} f_k(\mathcal{D}_k;\Theta^t)$ to PS instead. Both of the methods are mathematically equivalent [1], [12], [25], [26].

2.2 Personalized Federated Learning (PFL)

The learning objective of FL (i.e., Eq. (1)) minimizes the aggregate of individual loss functions (i.e., Eq. (2)) and derives generalized and common results for *all users using a single global model without any personalization*. Given the emergence of data heterogeneity across all users, directly optimizing average individual loss functions without personalization usually leads to unsatisfactory performance [11], [13], [14]. As a result, PFL instead considers a learning objective closer to each user. We refer to the setup in [13] to formalize PFL. Specifically, the loss function $f_n(\Theta)$ for each $n \in [N]$ changes to

$$f_n(\Theta) = \min_{\theta_n\in\mathbb{R}^d} f_n(\theta_n) + \frac{\lambda}{2}\|\theta_n - \Theta\|, \tag{5}$$

where θ_n is the personalized model of user n and λ is a regularization parameter that controls the strength of Θ to the personalized model θ_n. The rationale behind is to enable users to pursue their own models in different directions based on the global model Θ. As such, PFL can be formulated as a bi-level problem:

$$\min_{\Theta\in\mathbb{R}^d} \hat{f}(\Theta) \coloneqq \frac{1}{N}\sum_{n=1}^{N} g_n(\Theta),$$
$$\text{where } g_n(\Theta) = \min_{\theta_n\in\mathbb{R}^d} g_n(\theta_n) + \frac{\lambda}{2}\|\theta_n - \Theta\|. \tag{6}$$

For communication, PFL usually follows the communication scheme of FL (i.e., Eq. (4)) to update the global model Θ so there is no additional communication overhead [13].

2.3 Semi-supervised Learning (SSL)

Labeled data is considered difficult and costly to acquire in many cases, such as object detection in pictures or videos. SSL is an appropriate approach to resolve this difficulty. It assumes that a small amount of labeled data and a large amount of unlabeled data are available during training. Formally speaking, given a set of labeled data $x_1, ..., x_l \in X$ with corresponding labels $y_1, ..., y_l \in Y$ and a set of unlabeled data $x_{l+1}, ..., x_{l+u} \in X$, the goal of SSL is to infer a mapping from X to Y such that the labels Y are as closer as possible to the ground truth labels $\overline{Y}$ [27]. Existing work on SSL is divided into two main categories. One adds an unsupervised loss term (i.e., a regularizer) into the loss function so the training model is expected to learn the labeled and unlabeled data at the same time [28]–[31]; the other labels the unlabeled data with pseudo labels and the pseudo-labeled data are then used in training with a supervised loss [32]–[35].

2.4 Ensemble Learning

Ensemble learning is a general approach to improve predictive accuracy by combining the predictive results from multiple independent models. There are three major classes of ensemble learning methods: bagging, stacking, and boosting. Bagging fits multiple learning models with different training data and aggregates the predictive results by voting or averaging. Then, stacking that makes use of varying model types to fit on the training data and uses another model to combine the predictive results. Finally, boosting incrementally adds models that correct the predictive results made by the previous models and aggregates the final predictive results by weighted averaging [36]–[38].

3 System Model and Problem Formulation

In this section, we first elaborate the system model and background in Section 3.1, and then we introduce the dual-objective learning problem in Section 3.2.

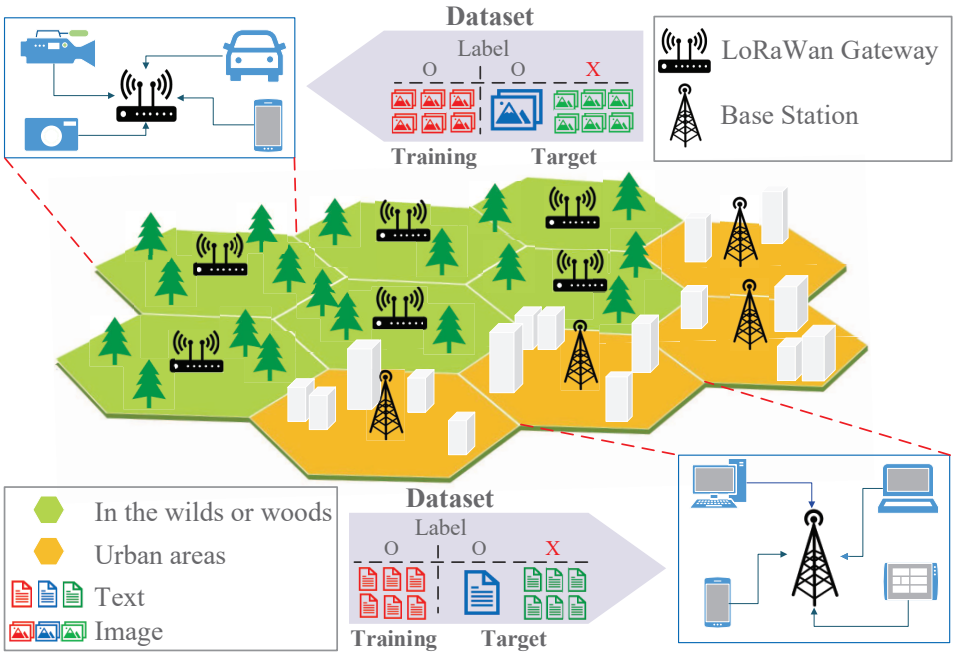


Fig. 1. We consider a scenario where a BS serves a set of devices. The devices each have two datasets, the training dataset and the target dataset. The data points in the training dataset are fully-labeled, whereas those in the target dataset are partially-labeled. Moreover, the number of labeled data points in the target dataset (i.e., the texts and images in blue) is the smallest among the labeled data points in the training dataset (i.e., the texts and images in red) and the unlabeled data points in the target dataset (i.e., the texts and images in green).

3.1 System Model

We consider a cellular network that consists of one BS serving a set of devices $[N]$. BS plays the role of PS and the set of devices are the users in FL as shown in Figure 1. In each communication round of FL, PS first broadcasts the global model to the users. Upon receiving the global model, users perform local training using their local data. When the local training is over, users upload the trained local models to PS for model aggregate.

The computational and transmission overhead and the data distributions of the devices are modeled as follows.

Computation on the Devices

Denote f_n as the computational capacity of device n. The computational capacity is usually measured by the number of CPU cycles per second. Let I_n, D_n, and C_n denote the number of local iterations at device n, the number of training data device n holds, and the number of CPU cycles for device n to traverse one data point (i.e., forward and backward propagation in local training), respectively. Therefore, the computation time on device n for local training is

$$t_n^l = \frac{I_n C_n D_n}{f_n}, \qquad \forall n \in [N]. \tag{7}$$

Transmission on the Devices

Recall that the devices should upload the local models to BS for the model aggregate. Following Shannon capacity, the transmission rate of device n is up to

$$r_n = b_n \log_2(1 + \frac{g_n p_n}{\mathcal{N} b_n}), \qquad \forall n \in [N], \tag{8}$$

where b_n is the bandwidth allocated to device n, g_n, the channel gain between device n and BS, p_n, average transmit power of device n, and $\mathcal{N}$, the Gaussian noise. Since the uploaded local models are usually dimensionally identical, we denote the local model size by m (e.g., the size of MobileNetV2 [39] is approximately 5MB). Therefore, the transmission time on device n to upload its local model is

$$t_n^u = \frac{m}{r_n}, \qquad \forall n \in [N]. \tag{9}$$

Computation and Transmission on BS

The role of BS in FL is responsible for the coordination of local models (computation and transmission). Since BS is usually equipped with high computational capacity, and the downlink bandwidth is much more sufficient than the uplink bandwidth, computational overhead and downlink transmission are assumed negligible compared to local training overhead and uplink transmission from devices [14], [25], [40], [41]. Therefore, we ignore the computational and model downlink transmission overhead at the BS side.

Data Distributions

We assume that each device n has a set of *training dataset* $\mathcal{D}_n^{train}$ and *target dataset* $\mathcal{D}_n^{target}$. In particular, the training dataset $\mathcal{D}_n^{train}$ consists of *the data that are all well-labeled and can be acquired easily by the user of the device* n. In contrast, the target dataset $\mathcal{D}_n^{target}$ is rather difficult for the user of device n to attain and correctly label. Each data point in the training dataset $\mathcal{D}_n^{train}$ is presented in the format of a feature-label pair (x, y). The features represent the data itself. For example, features in the image classification task are images that cover multiple objects. Labels represent the ground truth of the corresponding features. In the image classification task, for example, the labels show the classes of the objects in

the corresponding image. For the target dataset $\mathcal{D}_n^{target}$ on device n, we assume that *only a tiny amount of data points is labeled*, denoted as $\mathcal{D}_{n,l}^{target} \subset \mathcal{D}_n^{target}$. Furthermore, due to the difficulty in acquiring the target data, we assume that the number of the labeled data points in the target dataset is much smaller than that of the unlabeled data points in the target dataset, i.e., $|\mathcal{D}_{n,l}^{target}| << |\mathcal{D}_{n,ul}^{target}|$ for all N devices, where $|\cdot|$ denotes the cardinality and $\mathcal{D}_n^{target} = \mathcal{D}_{n,ul}^{target} \cup \mathcal{D}_{n,l}^{target}$.

3.2 Problem Formulation

Learning Objectives

Recall that each device owns two datasets, the training dataset $\mathcal{D}_n^{train}$ and the target dataset $\mathcal{D}_n^{target}$, where the target dataset consists of labeled data $\mathcal{D}_{n,l}^{target}$ and unlabeled data $\mathcal{D}_{n,ul}^{target}$, and $\mathcal{D}_{n,l}^{target} \cup \mathcal{D}_{n,ul}^{target} = \mathcal{D}_n^{target}$. The users of the devices participate in FL in order to satisfy their own two goals. First, *the users each wish to find an optimal model that best fits the labeled data in their target datasets as formally defined as follows.*

Definition 1 (*The personalized classification objective*)**.** The personalized classification objective demands an optimal model $\theta_{n,l}^{tar^*}$ that yields the minimum loss with respect to the labeled target data $\mathcal{D}_{n,l}^{target}$, i.e.,

$$\theta_{n,l}^{tar^*} = \underset{\theta_l^{tar} \in \mathbb{R}^d}{\arg\min}\, l_n^{cla}(\theta_l^{tar}; \mathcal{D}_{n,l}^{target}), \qquad \forall n \in [N], \tag{10}$$

where $l_n^{cla}(\cdot,\cdot)$ represents the loss function for classification on device n.

Note that the loss function depends on the application. It might be the cross entropy for image classification tasks (e.g., CIFAR10 [42]), or mean square error alternatively for natural language processing tasks (e.g., Spam Text Message Classification [43]). In addition to the personalized classification objective, *the users each also wish to find an optimal model that labels their unlabeled data $\mathcal{D}_{n,ul}^{target}$ as correctly as possible.*

Definition 2 (*The personalized labeling objective*)**.** The personalized labeling objective seeks an optimal model $\theta_{n,ul}^{tar^*}$ such that

$$\theta_{n,ul}^{tar^*} = \underset{\theta_{ul}^{tar} \in \mathbb{R}^d}{\arg\min}\, l_n^{lab}(\theta_{ul}^{tar}; \mathcal{D}_{n,ul}^{target}), \qquad \forall n \in [N], \tag{11}$$

where $l_n^{lab}(\cdot,\cdot)$ is the loss function for labeling on device n.

Remark. *The objectives Eq. (10) and Eq. (11) differ from the objectives of FL (i.e., Eq. (1)) and PFL (i.e., Eq. (6)) in two aspects. On the one hand, Eq. (10) asks for an optimal model for inferring the labeled target data, while Eq. (1) and Eq. (6) focus on the entire data (i.e., all the devices) and the entire personalized data (i.e., all the local data), respectively. On the other hand, Eq. (11) requires that a model label the unlabeled target data as accurately as possible without the aid of ground truth labels, while Eq. (1) and Eq. (6) assume that all the data used for training are well-labeled. Therefore, this paper studies a substantially more challenging problem than the existing work.*

Time Constraints

Based on Eqs. (7) and (9), it takes device n at least t_n^l to finish local training and at least t_n^u to upload its local model to BS in each communication round. Suppose that the number of total communication rounds is G. Then, the total training time for device n is at least

$$T_n = G(t_n^l + t_n^u), \qquad \forall n \in [N]. \tag{12}$$

Since the longer the training time is, the more the computational and transmission resources are used, the total training time is not allowed to exceed the predefined maximum training time T, that is,

$$T_n \leq T, \qquad \forall n \in [N]. \tag{13}$$

Remark that the above constraint on training time is more user-friendly, since users care more about how long it takes their devices to complete training over the specific amount of transmission and computational resources (i.e., bandwidth allocation and CPU cycles) [44], [45].

Finally, we formulate the **D**ual-**o**bjective **L**earning **P**roblem with partially-labeled data within the constraint of training service time (DoLP) as follows.

Definition 3. (DoLP) Given a set of devices $[N]$, where each device n owns two datasets, the training dataset $\mathcal{D}_n^{train}$ and the target dataset $\mathcal{D}_n^{target}$, total training time for device n T_n, the predefined maximum training time T, and the total communication round G, DoLP asks for two models for each device $n \in [N]$ that minimize the personalized classification objective:

$$\underset{\theta_l^{tar} \in \mathbb{R}^d}{\arg\min}\, l_n^{cla}(\theta_l^{tar}; \mathcal{D}_{n,l}^{target}), \qquad \forall n \in [N],$$

and the personalized labeling objective:

$$\underset{\theta_{ul}^{tar} \in \mathbb{R}^d}{\arg\min}\, l_n^{lab}(\theta_{ul}^{tar}; \mathcal{D}_{n,ul}^{target}), \qquad \forall n \in [N],$$

subject to the time constraint:

$$T_n \leq T, \qquad \forall n \in [N].$$

4 DoFed-SPP Design

In this section, we introduce **D**ual-**o**bjective **Fed**erated Learning System for the **S**ervices of **P**ersonalization and **P**artially-labeled Data (DoFed-SPP) design to solve the proposed DoLP. We first propose a complementary training scheme and then introduce the deliberately designed *similarity ratios* to measure the heterogeneity of the data between devices in Section 4.1. Then, Section 4.2 introduces the ensemble-based method for the personalized labeling objective. Third, an approach to addressing the personalized classification objective is presented in Section 4.3. Fourth, Section 4.4 addresses the optimization problem in DoFed-SPP. Lastly, the integration of *principle component analysis* (PCA) in DoFed-SPP for scalability is studied in Section 4.5.

4.1 Complementary Training Scheme

Reciprocal Model Training

In response to the first challenge of heterogeneous data distributions between the training dataset and the target dataset, we design a complementary training scheme in DoFed-SPP. Specifically, each device in DoFed-SPP performs local training according to the following objective.

$$\theta_n^{rec} = \arg\min_{\theta^{rec}\in\mathbb{R}^d} l_n(\theta^{rec}; \mathcal{D}_n^{train}), \qquad \forall n \in [N], \tag{14}$$

where it aims at training a *reciprocal model* θ_n^{rec} via

$$\theta_n^{rec} \leftarrow \theta_n^{rec} - \gamma \nabla_\theta l_n(\theta_n^{rec}; \mathcal{D}_n^{train}). \tag{15}$$

Local training in DoFed-SPP is equivalent to that in FL or PFL, where Eq. (15) leverages *stochastic gradient descent* (SGD) to attain a better reciprocal model (i.e., θ_n^{rec} in Eq. (15)) with the gradient (i.e., $\nabla_\theta l_n(\cdot;\cdot)$ in Eq. (15)). *The reciprocal models can best represent the data distributions of the training dataset on the devices*. Although the reciprocal model trained on one device may not be able to satisfy its own personalized classification and labeling objectives, it might be *complementary to other devices' two personalized objectives if the data distribution of the training dataset fits the target datasets of other devices.* The challenge here is that the complementary relations among the target dataset of devices and the reciprocal models are unknown, as the dataset are kept by the devices privately. Thus, we design an inference-based metric in DoFed-SPP to measure the similarity between all the devices as an indirect metric to estimate the complementary relations through SRs.

Inference-based Similarity Ratios

Recall that FL employs the weights w_k to average multiple local models in Eq. (4). The hidden assumption behind the model averaging (i.e., Eq. (4)) is that the users share a similar data distribution. If the assumption does not hold, the model averaging would not work properly [1], [22], [46]. To make the model averaging work in more general scenarios, such as heterogeneous data distributions on users' local data, it is indispensable to determine the optimal weights for aggregate by solving the following optimization problem.

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^N} f(\Theta), \tag{16}$$

where $\Theta = \sum_{n=1}^N w_n\theta_n$, and $\boldsymbol{w}^* = [w_1^*\ w_2^* ...\ w_N^*]^\top$ are the optimal weights for the model averaging (i.e., Eq. (4)). Following the same logic, *we derive the SRs for each device*. Denote by θ_n^{tar} the target model of device n. We aim to solve the optimization problem as follows to find the optimal SRs for each device n:

$$\boldsymbol{\rho}_n^* = \arg\min_{\boldsymbol{\rho_n}\in\mathbb{R}^N} l_n^{cla}(\theta^{tar}; \mathcal{D}_{n,l}^{target}), \qquad \forall n \in [N], \tag{17}$$

where $\theta^{tar} = \sum_{n=1}^N \rho_n^1\theta_n^{rec}$ and $\boldsymbol{\rho}_n^* \coloneqq [\rho_n^{1^*}\ \rho_n^{2^*} \dots \rho_n^{N^*}]^\top$ denotes the optimal similarity vector of the reciprocal model of device i on the labeled target data of device n. An intuitive method is iteratively solving Eq. (17) by leveraging SGD as follows to attain the optimal SRs:

$$\theta_n^{tar,t+1} = \theta_n^{tar,t} - \gamma \mathbf{1}^\top \nabla_{\boldsymbol{\rho}_n} l_n^{cla}(\theta_n^{tar,t}; \mathcal{D}_{n,l}^{target}), \tag{18}$$

where $\boldsymbol{\rho}_n \coloneqq [\rho_n^1\ \rho_n^2 \dots \rho_n^N]^\top$ and $\mathbf{1}^\top$ is a size-N vector of one. However, this straightforward method is infeasible in practice for the following two reasons: First, additional computational and transmission overhead for computing Eq. (18) among the devices is necessary to find the optimal $\boldsymbol{\rho}_n^*$. Second, the devices do not have sufficient labeled target data to learn the optimal SRs. Were it the case, *they could explicitly learn a target model by themselves rather than the SRs alone*, that is, no crucial need for them to participate in FL.

In light of the two reasons above, we approximate $\boldsymbol{\rho}_n^*$ with *first order approximation*. Before derivation, we require some assumptions about the analysis of gradient-based iterations. For notational convenience, let $\|\cdot\|$ denote l^2 norm in the following sections unless otherwise specified.

Assumption 1. *The maximum distance between $\theta, \theta' \in \mathbb{R}^d$ is bounded by a nonnegative constant G_1, that is, $\|\theta - \theta'\| \le G_1$.*

Assumption 2. *For every $n \in [N]$, l_n^{cla} is L-smooth, and its gradient is bounded by a nonnegative constant G_2, that is,*

$$\begin{aligned} &\|\nabla l_n^{cla}(\theta) - \nabla l_n^{cla}(\theta')\| \le L\|\theta - \theta'\|, \quad \forall \theta, \theta' \in \mathbb{R}^d, \\ &\|\nabla l_n^{cla}(\theta)\| \le G_2, \qquad\qquad\qquad\quad \forall \theta \in \mathbb{R}^d. \end{aligned}$$

The assumptions above are standard and typical for analyzing the iteration of gradient descent [47]–[54]. Remark that *we do not assume the convexity on l_n^{cla}* such that the SRs can be extended to more general cases.

Theorem 1. *Suppose that Assumptions 1 and 2 hold, the optimal SR $\rho_n^{i^*}$ of the reciprocal model of the device i on the labeled target data set of the device n, $\mathcal{D}_{n,l}^{target}$, can be approximated by $\hat{\rho}_n^i$, where*

$$\hat{\rho}_n^i = \frac{\max\left\{\frac{\gamma\mathcal{L}_{n,i}^{t+1} + G_1 + \gamma G_2}{\|\theta_i^{rec,t+1} - \theta_n^{rec,t}\|}, 0\right\}}{\sum_{j\in N}\max\left\{\frac{\gamma\mathcal{L}_{n,j}^{t+1} + G_1 + \gamma G_2}{\|\theta_j^{rec,t+1} - \theta_n^{rec,t}\|}, 0\right\}}, \tag{19}$$

where $\mathcal{L}_{n,i}^{t+1} \coloneqq l_n^{cla}(\theta_n^{tar,t}; \mathcal{D}_{n,l}^{target}) - l_n^{cla}(\theta_i^{rec,t+1}; \mathcal{D}_{n,l}^{target})$ is the target loss between the loss of the target model of device n on its labeled target data in communication round t and that of the reciprocal model of device i on the labeled target data of device n in communication round $t+1$.

Theorem 1 also holds for stochastic gradient descent (SGD) cases, and the proof can be found in Appendix A. The approximate $\hat{\rho}_n^i$ is time-varying by t and $t+1$, so the devices update their SRs in every communication round.

Remark. *The two terms in $\mathcal{L}_{n,i}^{t+1}$ represent how complementary the reciprocal model $\theta_i^{rec,t+1}$ can be. Specifically, if device n's target model $\theta_n^{tar,t}$ is insufficiently fit to his target dataset $\mathcal{D}_n^{target}$, then the first term in $\mathcal{L}_{n,i}^{t+1}$ will be large. Similarly, if device i's training data $\mathcal{D}_i^{train}$ is similar to $\mathcal{D}_n^{target}$, then the second term in $\mathcal{L}_{n,i}^{t+1}$ will be small. As a result, $\mathcal{L}_{n,i}^{t+1}$ will be large, implying that the reciprocal model of device i is complementary to the personalized objectives of device n so device n should weigh the reciprocal model of device i much more than other devices. Therefore, the physical meaning of SR is to grasp the importance of a reciprocal model for the labeled target data of another device.*

Local Training Warm-Up

The rationale behind the reciprocal model training and inference-based SRs is to find the relations between devices'

training dataset and target dataset. However, it is likely that the reciprocal models are not representative to their training data, leading to unrepresentative results of SRs. To this end, at initialization, the devices are required to perform *local training warm-up*.

Specifically, the devices should perform a given number of round L of local training using the objective in Eq. (14) and the SGD-based model update in Eq. (15) before uploading the reciprocal models. The local training warm-up can make the reciprocal models more representative to their training datasets, thereby leading to more stable and accurate SRs and superior final performance (see Appendix B for more details).

4.2 Personalized Labeling

Ensemble-based Pseudo-Labeling

Recall that each device establishes an approximated N-dimension similarity vector $\hat{\boldsymbol{\rho}}_n$ through Eq. (19), where the similarity vector helps the devices grasp how similar other devices' reciprocal models are to their labeled target data. Then, each device downloads a subset of reciprocal models $[B] = \{\theta_i^{rec}, \cdots, \theta_B^{rec}\}$ that score the B highest similarity ratios out of the set of available reciprocal models $[K]$ from BS, where $[B] \subseteq [K] \subseteq [N]$. We defer the decision of K and B to Section 4.4.

The rationale behind Personalized Labeling is that devices use multiple reciprocal models to achieve different opinions about unlabeled data points in the target datasets, and then weight the opinions with SRs. Thus, each device infers the unlabeled data points in the target dataset $\mathcal{D}_{n,ul}^{target}$ with the set of reciprocal models $[B]$ to predict the labels and averages the results with similarity ratios, that is,

$$\hat{p}(x) = OneHot\Big(\sum_{i\in[B]} \vec{p}(x;\theta_i^{rec}) \cdot \hat{\rho}_n^i\Big), \quad \forall x \in \mathcal{D}_{n,ul}^{target}, \tag{20}$$

where $OneHot(\cdot)$ denotes the one-hot function that rounds up the maximum element to 1 and rounds off the rest to 0, and $\hat{p}$, the pseudo label for the unlabeled data point x in the target dataset $\mathcal{D}_{n,ul}^{target}$. With Eq. (20), the devices can attain the pseudo labels for their unlabeled data points. The name *pseudo* is since the labels predicted by the reciprocal models are not ground truth.

Fundamental Difference in Personalized Labeling

This ensemble learning-based method has been shown to be effective and to avoid overfitting [23], [55]. However, there are two critical designs in DoFed-SPP that differ from the previous work. First, DoFed-SPP takes advantage of SRs to distinguish a set of the most complementary reciprocal models from the device pools for an individual device. Then, the opinions of the high-SR reciprocal models are weighted with the corresponding SRs such that the opinions from the data distributions more similar to the current device hold much more sway than other opinions over the final labels (i.e., $\hat{p}(x)$). It should be noted that devices can accurately achieve the labels of unlabeled target data without conducting SSL or FSSL, which solves the second challenge (i.e., training on the small and partially-labeled target dataset).

4.3 Personalized Classification

Recall that each device can label the unlabeled data points in the target dataset with the reciprocal models that present similar data distributions. Intuitively, the set of reciprocal models can also be used for personalized classification using ensemble-based techniques. However, the direct adoption of reciprocal models in an ensemble-based fashion cannot overcome the third challenge of unrepresentative training and testing performance, as the number of labeled target data is too small to reflect exact performance. Moreover, the pseudo labels are not helpful at all since the pseudo labels are the predictive results of the set of reciprocal models itself.

To this end, DoFed-SPP uses the teacher-student architecture for the personalized classification objective. The high-level idea is that the devices can use a larger and more sufficient dataset (i.e., the teacher or the pseudo-labeled dataset) to train a target model (i.e., the student) for their personalized classification objectives. Iteratively, when the pseudo-labeled dataset is closer to the ground truth, the target model can be stronger and stronger. As a result, the final target models achieve generalized performance and are free from the limit of scarcity of labeled target data, which means the challenge of unrepresentative training and testing performance is resolved.

Specifically, the teacher is the set of labeled data predicted by reciprocal models (i.e., $\hat{p}(x)$ in Eq. (20)), called *pseudo-labeled dataset* $\mathcal{D}_n^p$. Meanwhile, the student is an initial model at the beginning and will keep learning the teacher's decisions by performing local training using the pseudo-labeled dataset as the training set. Thus, the student's objective is

$$\theta_n^{tar*} = \underset{\theta^{tar}\in\mathbb{R}^d}{\arg\min}\, l_n(\theta^{tar};\mathcal{D}_n^p), \qquad \forall n \in [N], \tag{21}$$

and the student solves the above objective by iterating

$$\theta_n^{tar} \leftarrow \theta_n^{tar} - \alpha\nabla_\theta l_n(\theta_n^{tar};\mathcal{D}_n^p), \tag{22}$$

where α denotes the learning rate. Eq. (22) leverages SGD and the pseudo-labeled dataset to attain a better target model θ_n^{tar} iteratively.

4.4 Decision of K and B

Recall that each device requires a set of reciprocal models for Personalized Labeling (Section 4.2) and Personalized Classification (Section 4.3). The greater the size of $[K]$ becomes, the more the uplink transmission is necessary (i.e., uploading of reciprocal models from devices to BS). Similarly, the greater the size of $[B]$ becomes, the more the computational resource on devices is consumed (i.e., labeling in Personalized Labeling). Furthermore, the selection of composition of reciprocal models $[K]$ to upload is also critical as selecting the set of devices whose reciprocal models complement the most of other devices' objectives (i.e., Definitions 1 and 2) can maximize final predictive performance. Therefore, it is necessary to cherry-pick the beneficial reciprocal models for uploading and downloading so that the two objectives can be maximized within the constraint of transmission and computational resources.

The Size of $[K]$

We first study the size of $[K]$ (i.e., K). To explore the intrinsic nature of the decision of K, we assume that each device is equipped with an identical computational unit and has identical transmission conditions. Therefore, the number of reciprocal models downloaded by devices (i.e., B) for Personalized Labeling device is identical. Note that $B \leq K$ since only the uploaded reciprocal models can be downloaded and used for Personalized Labeling. Following the setup for the computation and transmission in Section 3.1, the computation time of one device in one communication round of DoFed-SPP can be formulated as follows.

$$t^l = \frac{I \cdot C_{tra} \cdot D}{f} + \frac{C_{SR}}{f} + \frac{B \cdot C_{pse}}{f}, \tag{23}$$

where C_{tra}, C_{SR}, C_{pse} denotes the number of CPU cycles for executing local training (i.e., training reciprocal models and teacher-student training), for calculating the SRs, and for inferring the unlabeled data points in $\mathcal{D}_{n,ul}^{target}$ with a reciprocal model (i.e., Personalized Labeling), respectively, and I, D, f represents the number of local iterations, the number of training data, and the computational capacity, respectively. x

For transmission time, we assume the total uplink transmission bandwidth is b and those K devices will share the uplink bandwidth equally. Then, the transmission time of uplink from K devices to BS is formulated as follows.

$$t^u = \frac{K \cdot m}{b \log_2(1 + \frac{g \cdot p}{\mathcal{N} b})}, \tag{24}$$

where m denotes the model size, g, the channel gain, p, the average transmit power, and $\mathcal{N}$, the Gaussian noise. Then, the total time of one device to execute one communication round of DoFed-SPP is

$$T_r = t^l + t^u \tag{25}$$

Recall that the total training time should not be greater than T. Suppose that the number of total communication rounds is G, and thus the training time of each communication round should not be greater than T/G, so we can obtain the following inequality.

$$\begin{aligned} \frac{T}{G} \geq T_r &= t^l + t^u \\ &= \frac{I \cdot C_{tra} \cdot D}{f} + \frac{C_{SR}}{f} \\ &\quad + \frac{B \cdot C_{pse}}{f} + \frac{K \cdot m}{b \log_2(1 + \frac{g \cdot p}{\mathcal{N} b})} \end{aligned}$$

By rearranging the term on the left-hand side, we have the following.

$$\frac{T}{G} - \frac{I \cdot C_{tra} \cdot D + C_{SR}}{f} \geq \frac{B \cdot C_{pse}}{f} + \frac{K \cdot m}{b \log_2(1 + \frac{gp}{\mathcal{N} b})} \tag{26}$$

It is worth noting that the first term and the second term on the right-hand side of Eq. (26) represent how much time remains for computation and transmission, respectively. Thus, we can examine every combination of B and K, according to Eq. (26), where $1 \leq B \leq K \leq N$, to find one combination of B and K that maximizes the expected performance since all the variables other than B, K are already known[1]. Specifically, let

$$C_1 = \frac{T}{G} - \frac{I \cdot C_{tra} \cdot D + C_{SR}}{f},$$
$$C_2 = \frac{C_{pse}}{f}, \text{ and } C_3 = \frac{m}{b \log_2(1 + \frac{gp}{\mathcal{N} b})}.$$

It suffices to examine the following combinations of B and K, where $\max\{1, \lfloor \frac{C_1}{C_2 + C_3} \rfloor\} \leq K \leq \min\{N, \lfloor \frac{C_1 - C_2}{C3} \rfloor\}$ and $B = \min\{K, \lfloor \frac{C_1 - K \cdot C_3}{C_2} \rfloor\}$. Next, we formulate the expected performance for a given specific combination of B and K.

Algorithm 1 The $(1 - \frac{1}{e})$ Approximation Algorithm

Input: Given parameters: B, K, $\hat{\boldsymbol{\rho}} = [\hat{\boldsymbol{\rho}}_1\ \hat{\boldsymbol{\rho}}_2 \cdots \hat{\boldsymbol{\rho}}_N]$, and a set of devices $[N]$.
Output: The set of selected devices $[K]_B$

```
[K]_B ← ∅;
while |[K]_B| < K do
    n ← arg max_{n∈[N]} ρ̂([K]_B ∪ {n}) − ρ̂([K]_B); ▷ ρ̂([K]_B) denotes the total similarity ratios of the set of reciprocal models [K]_B (see Definition 5 in Appendix C).
    [K]_B ← [K]_B ∪ {n};
    N ← N \ {n};
return [K]_B
```

The Set for a Combination B and K

Given a specific combination of B and K, we now turn to another problem that *which K reciprocal models should be selected such that all the devices can expect to have maximum performance on their two personalized objectives*. For ease of reading, denote by $[K]_B$ the set $[K]$ given with B. Based on the previous work regarding ensemble learning, the more the models trained with similar data distributions come to serve, the better the final performance can be [5], [58] (See Section 6 for more details). Therefore, the optimization problem of the set $[K]_B$ is defined as follows.

Definition 4 (*The optimization problem of the set* $[K]_B$ **).** Given the parameters, B, K, $\hat{\boldsymbol{\rho}} = [\hat{\boldsymbol{\rho}}_1\ \hat{\boldsymbol{\rho}}_2 \cdots \hat{\boldsymbol{\rho}}_N]$, and a set of devices $[N]$, the optimization problem of the set $[K]_B$ asks for a set of reciprocal models such that

$$\underset{\substack{x_{ni}, y_i \in \{0,1\}, \\ \forall n, i \in [N]}}{\text{maximize}} \quad \sum_{n \in [N]} \sum_{i \in [N]} x_{ni} \cdot \hat{\rho}_n^i \tag{27}$$

$$\text{subject to} \quad \sum_{i \in [N]} x_{ni} \leq B, \qquad \forall n \in [N], \tag{28}$$

$$\sum_{i \in [N]} y_i \leq K, \tag{29}$$

$$x_{ni} \leq y_i, \qquad \forall n, i \in [N], \tag{30}$$

$$x_{ni}, y_i \in \{0, 1\}, \qquad \forall n, i \in [N], \tag{31}$$

where x_{ni} and y_i are two decision variables that denote whether device n uses the reciprocal model of device i for Personalized Labeling and whether the reciprocal model of device i is uploaded to BS, respectively.

The objective (27) maximizes the sum of the SRs over all the devices. Two constraints (28) and (29), derived from

1. It is reasonable to assume that BS is able to get hold of the computational capability and transmission conditions of the devices with which it associates [56], [57].

Ineq. (26), limit the total numbers of the reciprocal models for Personalized Labeling and uploading to B and K, respectively. The third constraint (30) indicates that only the uploaded reciprocal models can be used. Finally, the last constraint (31) indicates two decision variables x_{ni} and y_i.

Obviously, the optimization problem of the set $[K]_B$ is NP-hard since it is a variant of the weighted maximum coverage problem [59]. To this end, we design an approximation algorithm (i.e., Algorithm 1) and show the approximation ratio in Theorem 2.

Theorem 2. *Algorithm 1 achieves an approximation ratio of* $1 - \frac{1}{e} \approx 0.632$.

Please refer to Appendix C for the proof of Theorem 2. In each communication round, DoFed-SPP calls Algorithm 1 to calculate $[K]_B$ for each possible combination of B and K and then determine which K reciprocal models should be uploaded so that the expected training performance on all devices can be maximized while adhering to time constraints. That is, the selected $[K]$ has the maximum sum of the SRs among all sets $[K]_B$ of the different combinations of B and K.

4.5 Scalability with respect to Large Models

In light of the varying requirements on *machine learning* (ML)-based applications, distinct models that have different sizes and number of parameters are used. For example, *convolutional neural networks* (CNNs) that have different number of layers and structures aim at image classification tasks, and *bidirectional encoder representations from transformers* (BERT) and its variants for *natural language processing* (NLP) tasks. The model sizes may range from 5MB (e.g., MobileNet [39]) to 100MB (e.g., BERT [60], [61]). The larger the model size, the larger the extra computational and transmission overhead in DoFed-SPP.

To this end, DoFed-SPP leverages PCA [62] to downsize the dimensions of the given models. PCA is shown to be effective and barely compromises the testing performance [25], [62] if the reduced size is well defined empirically (refer to Section 6 for more details). In practice, BS runs PCA to downsize the dimensions of all the reciprocal models before broadcasting them back to the devices. The advantages are two-fold. On the one hand, the devices can consume less computational and memory resources to calculate SRs (Section 4.1) and execute inference (Section 4.2) because the dimensions of the reciprocal models are reduced. On the other hand, no additional computational overhead on the devices for executing PCA is generated since BS finishes the execution of PCA first, and then broadcasts the reduced reciprocal models to the devices.

5 Workflow of DoFed-SPP

In this section, we summarize all the phases in DoFed-SPP. The system diagram of DoFed-SPP is shown as Fig. 2. Suppose that there is a set of devices $[N]$ that participate the training and a BS that coordinates the wireless transmission. For each phase in DoFed-SPP, the descriptions are as follows.

Phase 0: **Cold Start.** All devices $n \in [N]$ first initialize models in Phase 0-1, then perform L rounds of local training to establish reciprocal models θ_n^{rec} in Phase 0-2, and finally upload the reciprocal models to BS in Phase 0-3. BS first calls PCA to downsize the dimensions of the reciprocal models in Phase 0-4[2] and broadcasts all of them to the devices in

Algorithm 2 DoFed-SPP

Output: Target model $\theta_n^{tar,t}$ for the personalized classification objective, and a set of downsized reciprocal models $\hat{\mathcal{S}}_t$ for the personalized labeling objective.

Phase 0: ***Cold Start***

$t \leftarrow 0$;
for each device $n \in [N]$ *runs in parallel* **do**
  $\theta_n^{rec,0}, \theta_n^{tar,0} \leftarrow$ `model_init()`;
  $\theta_n^{rec,1} \leftarrow$ `local_training_warmup`$(\theta_n^{rec,0}, \mathcal{D}_n^{train}, L)$;
  `upload_to_BS`$(\theta_n^{rec,1})$;
for each model $\theta_n^{rec,1}$, *BS runs in sequential* **do**
  $\hat{\theta}_n^{rec,1} \leftarrow$ `sklearn.decomposition.PCA`$(\theta_n^{rec,1})$;
  `broadcast_to_dev`$(\hat{\theta}_n^{rec})$;
for each device $n \in [N]$ *runs in parallel* **do**
  $\hat{\boldsymbol{\rho}}_n \leftarrow$ `Calculate_SR`$(\hat{\boldsymbol{\theta}}^{rec,1}, \theta_n^{tar,0}, \theta_n^{rec,0}, \mathcal{D}_{n,l}^{target})$; ▷ run Eq. (19) in Theorem 1.
  `upload_to_BS`$(\hat{\boldsymbol{\rho}}_n)$;
$t \leftarrow t + 1$;

Phase 1: ***Decision of*** K *(All the statements run by the BS)*

All combinations of $B, K \leftarrow$ `value_of_B_K()`; ▷ run Ineq. (26).
$[K]_B \leftarrow$ `Algorithm_1`$(B, K, [\hat{\boldsymbol{\rho}}_{\boldsymbol{n}}]_{n\in[N]})$ for each combination of B and K;
$[K]_t \leftarrow \arg\max_{[K]_B} \rho([K]_B)$;
$[\hat{K}]_t \leftarrow \emptyset$; ▷ the set of downsized reciprocal models.
if $t > 1$ **then**
▷ PCA should be called again if not at the first round
  for each $\theta_n^{rec,t} \in [\hat{K}]_t$ **do**
    $\hat{\theta}_n^{rec,t} \leftarrow$ `sklearn.decomposition.PCA`$(\theta_n^{rec,t})$;
    $[\hat{K}]_t \leftarrow [\hat{K}]_t \cup \{\hat{\theta}_n^{rec,t}\}$;
for each $\hat{\theta}_n^{rec,t} \in [\hat{K}]_t$ **do**
  `broadcast_to_dev`$(\hat{\theta}_n^{rec,t})$;

Phase 2: ***Personalized Labeling***

for each device $n \in [N]$ *runs in parallel* **do**
  $\mathcal{D}_n^p \leftarrow \emptyset$; ▷ the pseudo-labeled dataset.
  for each $x \in \mathcal{D}_{n,ul}^{target}$ **do**
    $\hat{p}(x) \leftarrow OneHot\Big(\sum_{i\in[\hat{K}]_t} \vec{p}(x; \hat{\theta}_i^{rec,t}) \cdot \hat{\rho}_n^i\Big)$ ▷ run Eq. (20).
    $\mathcal{D}_n^p \leftarrow \mathcal{D}_n^p \cup \{(x, \hat{p}(x))\}$;
  $\mathcal{D}_n^p \leftarrow \mathcal{D}_n^p \cup \mathcal{D}_{n,l}^{target}$ ▷ combine the ground truth labeled target data.

Phase 3: ***Personalized Classification***

for each device $n \in [N]$ *runs in parallel* **do**
  $\theta_n^{tar,t+1} \leftarrow \theta_n^{tar,t} - \alpha \nabla_\theta l_n(\theta_n^{tar,t}; \mathcal{D}_n^p)$; ▷ run Eq. (22)

Phase 4: ***SRs and Reciprocal Model Update***

for each device $n \in [N]$ *runs in parallel* **do**
  $\hat{\boldsymbol{\rho}}_n \leftarrow$ `Calculate_SR`$(\hat{\boldsymbol{\theta}}^{rec,t}, \theta_n^{tar,t-1}, \theta_n^{rec,t-1}, \mathcal{D}_{n,l}^{target})$; ▷ run Eq. (19) in Theorem 1.
  $\theta_n^{rec,t+1} \leftarrow \theta_n^{rec,t} - \alpha \nabla_\theta l_n(\theta_n^{rec,t}; \mathcal{D}_n^{train})$; ▷ run Eq. (15)
  `upload_to_BS`$(\hat{\boldsymbol{\rho}}_n)$;
$t \leftarrow t + 1$;

2. The reciprocal models downsized by PCA can infer data as well. Please refer to Section 6 for more details.

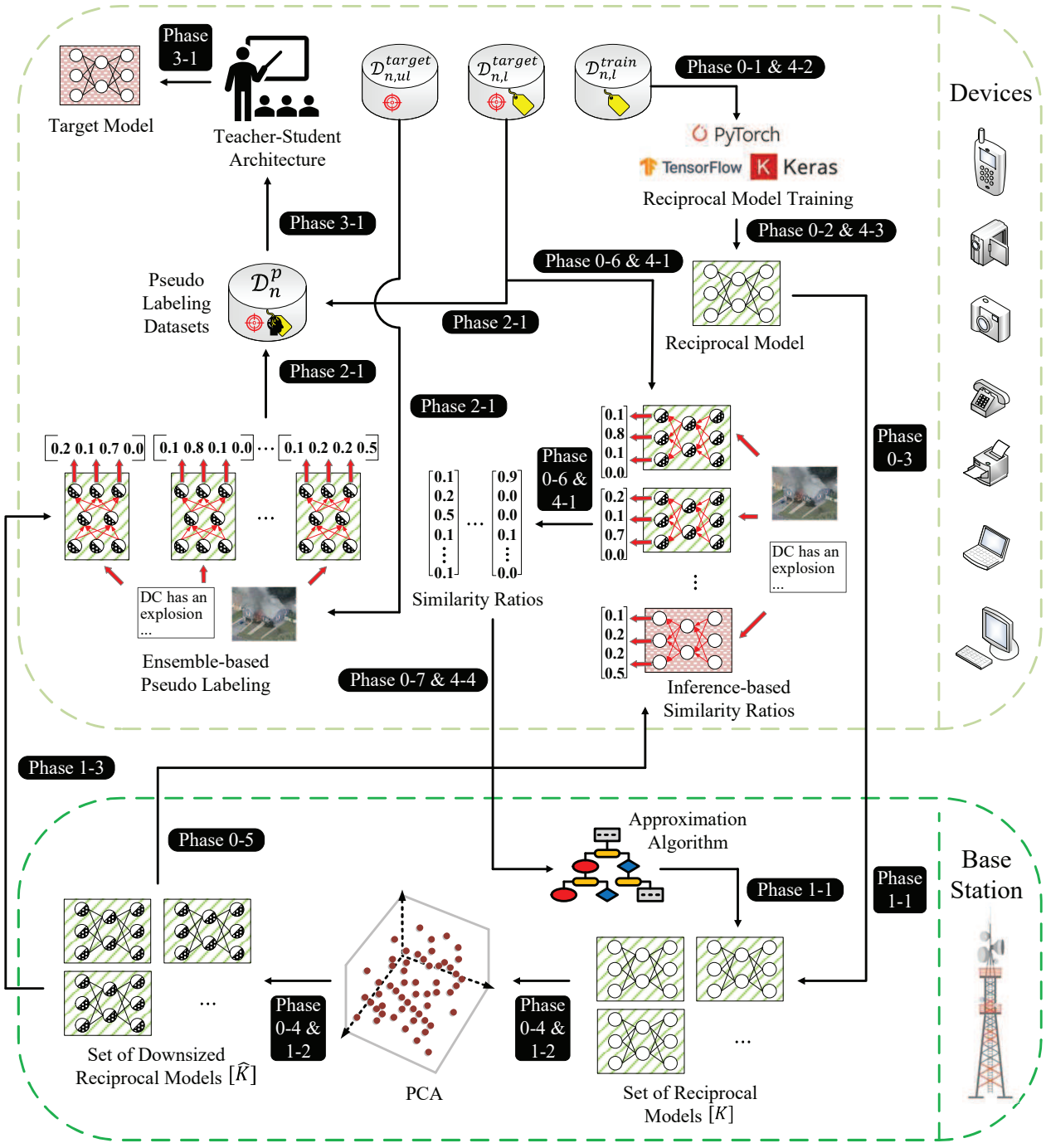


Fig. 2. The system diagram of DoFed-SPP.

Phase 0-5. The devices calculate SRs in Phase 0-6 (i.e., Eq. (19)) and send SRs back to BS in Phase 0-7.[3]

Phase 1: **Decision of K and B.** BS examines each combination of B and K by Eq. (26) , by running Algorithm 1, to calculate the corresponding set of devices $[K]_B$ and then determine the optimal set of devices $[K]$ that should upload their reciprocal models and the set of reciprocal models for each device (i.e., the solution with the decision variables y_i and x_{ni} that maximizes the sum of the SRs) in Phase 1-1. Then, the BS runs PCA for the reciprocal models uploaded from the set of devices $[K]$ in Phase 1-2, and transmits the set of reciprocal models downsized, denoted by $[\hat{K}]$, to all devices in Phase 1-3.

Phase 2: **Personalized Labeling.** When receiving the set of reciprocal models downsized $[\hat{K}]$, the devices use the subset of reciprocal models received $[\hat{B}] \subseteq [\hat{K}]$ to execute Eq. (20) so that the unlabeled data points in $\mathcal{D}_{n,ul}^{target}$ can be pseudo-labeled, and the pseudo-labeled data points combines with the ground truth data points in $\mathcal{D}_{n,l}^{target}$ to generate the pseudo-labeled dataset $\mathcal{D}_n^p$ in Phase 2-1.

Phase 3: **Personalized Classification.** After establishing the pseudo-labeled datasets, the devices use the teacher-

3. Since SRs are a N by N vector with elements ranging from $[0, 1]$, we omit the communication overhead of SRs from devices to BS.

student architecture to update the target model θ_n^{tar} for personalized classification (i.e., to follow the objective in Eq. (21) and the iterative process in Eq. (22)) in Phase 3-1.

Phase 4: **SRs and Reciprocal Model Update.** The devices update SRs of all other devices with the set of reciprocal models downsized $[\hat{K}]$ in Phase 4-1 and perform a one-round local training to update their own reciprocal models in Phase 4-2 to update reciprocal models in Phase 4-3. Devices can use the cached reciprocal models to calculate SRs if they are not involved in the current set $[\hat{B}]$. Finally, the devices upload the renewed SRs to BS in Phase 4-4.

Phases 1 to 4 will repeat until the given stopping criteria are met (please refer to Section 6.1 for more details). Remark that the accuracy of the pseudo-labeled dataset in phase 2 will be improved iteration by iteration since the reciprocal models will be updated in phase 4, which also helps enhance the personalized classification models in phase 3. The pseudocode of DoFed-SPP is presented in Algorithm 2.

6 Performance Evaluation

In this section, we conduct extensive experiments on DoFed-SPP. Subsection 6.1 details the setup of the experiments. The performance of DoFed-SPP and the four baselines using two performance metrics is presented in Subsections 6.2 and 6.3, respectively. Due to the page limit, we perform the sensitivity analysis of DoFed-SPP in Appendix B to view the impact of several parameters in DoFed-SPP.

6.1 Implementation Setup

Datasets and Default Model

Three datasets, CIFAR10 (10 classes), CIFAR100 (100 classes) [42], and DBPedia (14 classes) [63], are used for evaluating the performance of DoFed-SPP and other baselines. CIFAR10 and CIFAR100 are two image classification datasets, whereas DBPedia is a text classification dataset. A model composed of two `nn.conv2d` layers and two `nn.linear` layers serves as the default model for the two image classification datasets, while the model composed of one `nn.EmbeddingBag` layer plus one `nn.Linear` layer [64] is used as the default model for the text classification dataset. The sizes of the two models are approximately 2.1MB and 51MB, respectively. For the hyperparameters setup and model detail, please refer to Appendix D.

Data Distributions

We refer to [12], [14] to split the data of the three datasets in two ways. The first one is *SUBSET*. In SUBSET, we split all classes of data into $\mathcal{K} \in \mathbb{Z}^+$ clusters at random. If the number is not divisible by $\mathcal{K}$, then the remaining classes are assigned to any cluster at random. Then, each device $n \in [N]$ is assigned a set of training dataset $\mathcal{D}_n^{train}$ and a set of target dataset $\mathcal{D}_n^{target}$, both of which are drawn from two distinct clusters, where $\mathcal{D}_n^{train} = 500 \cdot \frac{C}{\mathcal{K}}$, $\mathcal{D}_n^{target} = \mathcal{D}_{n,l}^{target} \cup \mathcal{D}_{n,ul}^{target}$, $|\mathcal{D}_{n,l}^{target}| = 10 \cdot \frac{C}{\mathcal{K}}$, $|\mathcal{D}_{n,ul}^{target}| = 190 \cdot \frac{C}{\mathcal{K}}$, where C is the number of all classes. For example, say $\mathcal{K} = 5$ in CIFAR10, then each device n has only $\frac{10}{5} = 2$ classes in its training dataset and two other classes in its target dataset, and the number of training data points is $500 \cdot \frac{10}{5} = 1000$, the number of labeled and unlabeled target data points is $10 \cdot \frac{10}{5} = 20$ and $190 \cdot \frac{10}{5} = 380$, respectively. Obviously, the value of $\mathcal{K}$ represents the degree of data heterogeneity among the devices.

The second one is *NORMAL*. In NORMAL, each device is assigned a random number of classes in its training dataset and its target dataset, while only the assumption that the classes in the training dataset and those in the target dataset of one device do not completely overlap is made. Specifically, the numbers of classes in the training dataset and the target dataset of a device are drawn randomly in a range from 2 to 10. If the classes in the target dataset perfectly align with those in the training dataset, the classes for the two datasets are redrawn again. For instance, a device may be assigned class No.1 to No.2 in its training dataset but class No.2 to No.5 in its target dataset. The numbers of data points in the training dataset and the target dataset in NORMAL are similar to those in SUBSET.

Baselines

We compare DoFed-SPP with following four baselines: *Federated Averaging* (FedAvg) [1], FedFomo [14], *Federated Bayesian Ensemble* (FedBE) [5], and Local, all of which are implemented in PyTorch. More specifically, FedAvg is a naïve FL framework. FedFomo asks each device to learn a personalized model by averaging local models from other devices with specially designed weights. FedBE combines Bayesian model ensemble with FL in order to overcome the challenge of heterogeneous data. Local performs local training all the time without any model exchange. For more implementational details regarding DoFed-SPP and the baselines, please see Appendix E. The experimental results are averaged over *10 trials*.

Reference Training Environment

We refer to the setup in [1], [14], where there are 25 devices that wish to participate the training for their individual personalized classification objective and personalized labeling objective (i.e., $N = 25$). In each communication round, only a subset of devices, $[K]$, can upload the local models to the BS, where the default size of $[K]$ for FedAvg, FedFomo and FedBE is 10. For DoFed-SPP, the computation time is measured on an RTX3090 GPU, whereas the transmission time is a random variable from the transmission rate between 2MB/s to 10MB/s for uploading a single reciprocal model. For fair comparison, $K = B = 10$ for DoFed-SPP.

Performance Metrics and Stopping Criterion

We refer to two metrics to measure performance. One is *final accuracy* (FA) and the other is *time-to-accuracy performance* (TTAP). For FA, the final *top-1* mean testing accuracy over all the devices' target models on their target data is presented. For TTAP, the wall clock time (i.e., computation and transmission time) to train a model to achieve the top-1 accuracy over all target models of devices on their target data is considered.

The stopping criterion for FA is that the top-1 mean testing accuracy between two consecutive communication round is less than 0.01. The stopping criterion for TTAP is that the training process is shut down when the maximum wall clock time is up. The maximum wall clock time for

TABLE 1
Final Accuracy (classification accuracy / labeling accuracy)

	CIFAR10-SUBSET	CIFAR100-SUBSET	DBPedia-SUBSET
FedBE [5]	$0.56 \pm 0.07/0.58 \pm 0.08$	$0.33 \pm 0.15/0.35 \pm 0.05$	$0.46 \pm 0.02/0.45 \pm 0.04$
FedFomo [14]	$0.68 \pm 0.11/0.69 \pm 0.06$	$0.55 \pm 0.08/0.55 \pm 0.10$	$0.61 \pm 0.11/0.62 \pm 0.05$
FedAvg [1]	$0.18 \pm 0.11/0.19 \pm 0.09$	$0.02 \pm 0.03/0.02 \pm 0.01$	$0.08 \pm 0.05/0.10 \pm 0.01$
Local	$0.15 \pm 0.04/0.16 \pm 0.10$	$0.07 \pm 0.08/0.08 \pm 0.04$	$0.04 \pm 0.07/0.07 \pm 0.05$
DoFed-SPP	$\mathbf{0.89 \pm 0.03/0.94 \pm 0.02}$	$\mathbf{0.71 \pm 0.03/0.81 \pm 0.02}$	$\mathbf{0.80 \pm 0.11/0.84 \pm 0.05}$
	CIFAR10-NORMAL	CIFAR100-NORMAL	DBPedia-NORMAL
FedBE [5]	$0.58 \pm 0.12/0.60 \pm 0.09$	$0.44 \pm 0.15/0.45 \pm 0.07$	$0.61 \pm 0.21/0.63 \pm 0.12$
FedFomo [14]	$0.74 \pm 0.15/0.76 \pm 0.12$	$0.54 \pm 0.11/0.54 \pm 0.08$	$0.63 \pm 0.09/0.62 \pm 0.02$
FedAvg [1]	$0.40 \pm 0.25/0.42 \pm 0.04$	$0.36 \pm 0.14/0.38 \pm 0.09$	$0.33 \pm 0.05/0.32 \pm 0.10$
Local	$0.11 \pm 0.02/0.12 \pm 0.10$	$0.04 \pm 0.02/0.04 \pm 0.01$	$0.05 \pm 0.01/0.07 \pm 0.01$
DoFed-SPP	$\mathbf{0.82 \pm 0.06/0.87 \pm 0.07}$	$\mathbf{0.68 \pm 0.04/0.85 \pm 0.05}$	$\mathbf{0.71 \pm 0.06/0.85 \pm 0.05}$

CIFAR10/100 and DBPedia is 250 sec and 21,000 sec, respectively.

6.2 Final Accuracy

The FA of DoFed-SPP and the baselines are summarized in Table 1, whose field shows the *classification accuracy* and *labeling accuracy* on three datasets using two data distributions (i.e., SUBSET and NORMAL). It is obvious to see that DoFed-SPP significantly outperforms other methods regardless of datasets or data distributions. Remark that the performance of Local and FedAvg is rather awful, which implies that the naive local training without any collaboration and naive FL cannot satisfy DoLP directly.

Furthermore, we can see that for DoFed-SPP the labeling accuracy is regularly no lower than the classification accuracy. It is because DoFed-SPP adopts the teacher-student architecture, where the teacher is the set of reciprocal models downloaded while the student is a single model that learns the behavior of the teacher. However, the four baselines do not consider the personalized labeling objective so they use the models for classification to infer the personalized labeling tasks and the classification accuracy is similar to the labeling accuracy. Due to the page limit, more experimental results of labeling accuracy are in Appendix F.

6.3 Time-to-accuracy Performance

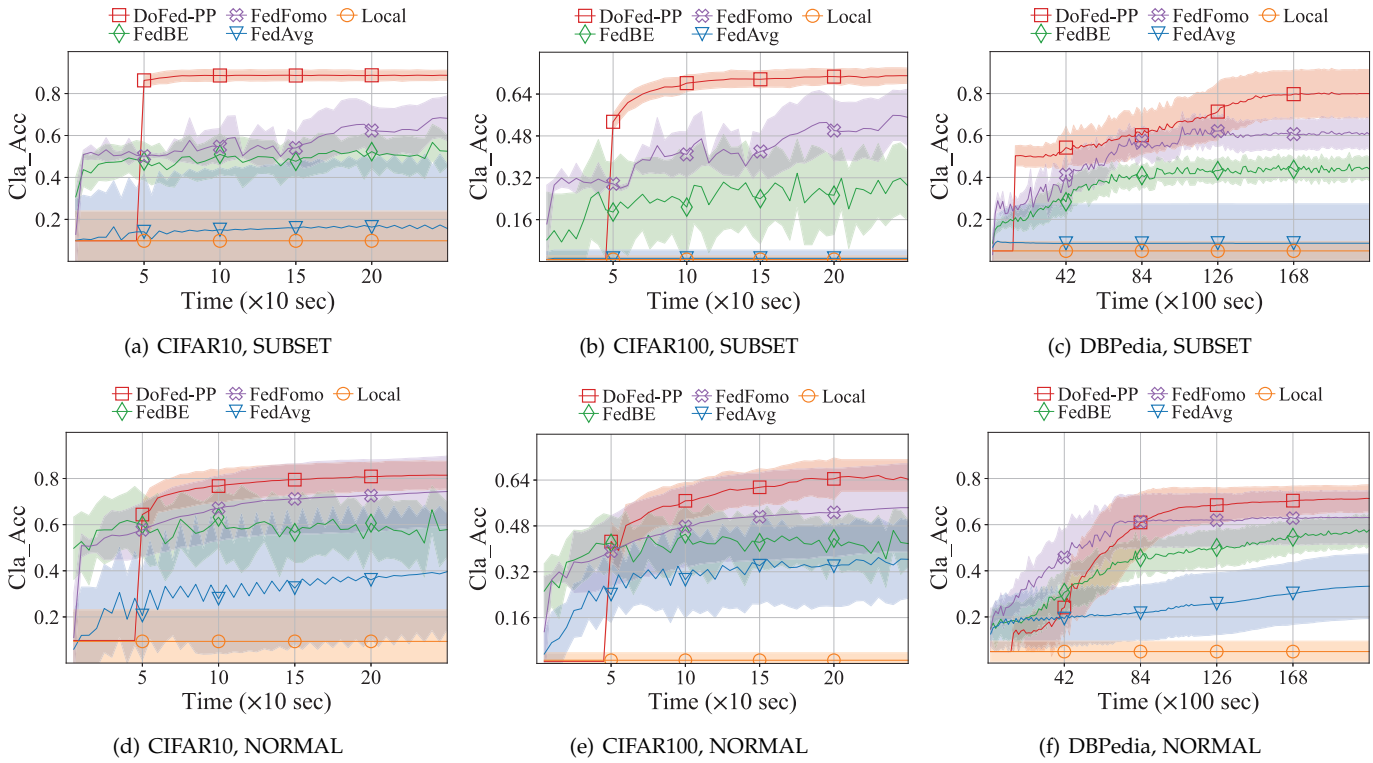
Figure 3 reports the TTAP of DoFed-SPP and the baselines in the three datasets using two data distributions, SUBSET and NORMAL. Clearly, DoFed-SPP stills outperforms the four baselines regardless of datasets and data distributions. Remark that DoFed-SPP performs the same as Local initially since DoFed-SPP initializes the local training warm-up on reciprocal models without any mode exchange. The accuracy of DoFed-SPP after the local training warm-up (i.e., Phase 0) skyrockets. It is because reciprocal models are sufficiently representative to their training datasets and using SRs the devices can find the most beneficial set of reciprocal models to satisfy personalized objectives. Moreover, the accuracy after the local training warm-up is ever-increasing since the teachers (i.e., the pseudo-labeled datasets) are enhancing the labeling accuracy, thereby improving the student's classification accuracy accordingly (i.e., the classification model).

7 Related Work

Existing studies can be divided into three categories: *Communication efficiency for FL*, *federated semi-supervised learning*, and *personalized federated learning*.

Communication efficiency for FL. Since training FL usually consumes a substantial deal of communication resources, much work attempts to improve the communication efficiency in training FL. Konečnỳ *et al.* proposed to use smaller number of variables to represent the local models from the participants so that the communication complexity between PS and the participants can be reduced [65]. Bonawitz *et al.* viewed the training of FL from the perspective of a top-down system and proposed a training mechanism to improve the communication and operation efficiency [66]. Han *et al.* sparsified the gradients from the participants with different degrees adaptive to participants' local data and strike a balance among the training performance, communication efficiency, and degrees of gradient sparficiation [67]. Huang *et al.* presented a novel encoding scheme for communication of FL between the participants and PS so that the communication efficiency can be further improved [68]. Jin *et al.* proposed an online participant selection algorithm for FL in favor of the dynamic scenarios where available participants are time-varying [48]. Wang *et al.* designed a three-layer (i.e., among participants and between PS and participants) communication hierarchy and combined the asynchronous updating technique with FL on to mitigate communication overhead derived from straggler effect [47]. Li *et al.* developed a flexible communication compression scheme guided by the convergence bound of FL and adaptive to the computing and communication conditions across the participants [51]. The work above focuses on the minimization on the average loss and do not consider the unlabeled data points, making them unsuitable for DoLP.

Federated semi-supervised learning (FSSL). Zhang *et al.* incorporated graph normalization into FSSL such that the gradient diversity from the different users can be mitigated, thereby improving testing accuracy [69]. Che *et al.* proposed FedTriNet, an FSSL framework that exploits the subtly-designed labeling mechanism to augment the insufficient amount of labeled data such that the test accuracy can be improved [70]. Kang *et al.* presented FedCVT to improve the testing accuracy when the amount of labeled data is insufficient. FedCVT estimates representations for missing features, predicts labels for unlabeled data to expand the



(a) CIFAR10, SUBSET (b) CIFAR100, SUBSET (c) DBPedia, SUBSET

(d) CIFAR10, NORMAL (e) CIFAR100, NORMAL (f) DBPedia, NORMAL

Fig. 3. TTAP of DoFed-SPP and the baselines on CIFAR10, CIFAR100, and DBPedia using two data distributions, SUBSET and NORMAL.

training set, and trains three classifiers jointly based upon different views of the expanded training set to improve the test accuracy [71]. Jeong *et al.* proposed an FSSL framework named FedMatch that minimizes the inter-client consistency loss for mitigating model diversity among the clients such that the test accuracy can be improved, and decomposes parameters into one for labeled data and the other for unlabeled data in order to reduce the training overhead [20]. However, the above work aims to minimize the loss of the aggregate of local models, that is, they do not consider the personalization of the users. Therefore, they are unable to address DoLP.

Personalized federated learning (PFL). Smith *et al.* proposed MOCHA, a multi-task learning for FL that considers the users as tasks and each user learns one model [72]. Fallah *et al.* employed a meta-learning approach to realize fast personalization, where the users can be easily adaptive to their local data by executing a few steps of gradient descent with respect to their local data [11]. Dinh *et al.* used Moreau envelopes as the users' regularized loss functions so that the personalized model optimization can be decoupled from the global model training for personalization [13]. Ghosh *et al.* proposed to cluster the users according to their local data distributions. The users in the same cluster update a global model such that the data heterogeneity can be mitigated [54]. Collins *et al.* proposed that the central server aggregates the representation layers of local models while the users keep unique heads of local models for personalization [73]. It is obvious to see that the aforementioned work assumes that the users' personalized objectives align with their local data, thereby unable to address DoLP.

8 Conclusions

In this paper, we make the first attempt to study the issue of insufficient and partially-labeled data in PFL, based on which we formulate the problem DoLP that has two personalized service objectives and the constraint of training time over wireless networks. Then, we propose a PFL service system DoFed-SPP that 1) adopts an inference-based first order approximation metric to determine the similarity between the local data of devices, and 2) uses an approximation algorithm to select the optimal size and set of devices to upload their local models. The extensive experiments show that DoFed-SPP outperforms the state of the art in terms of two performance metrics on three benchmarks. Due to the page limit, we defer the discussion and future work to Appendix G.

Acknowledgments

This work was supported in part by the National Science and Technology Council under Grant 108-2221-E-194-025-MY3, 108-2628-E-001-003-MY3, 111-2628-E-001-002-MY3, 111-2628-E-194-001-MY3, and 111-3114-E-194-001-, the Academia Sinica under Thematic Research Grant AS-TP-110-M07-2; and in part by the Advanced Institute of Manufacturing with High-tech Innovations (AIM-HI) from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. PMLR AISTATS*, 2017.

[2] B. Nour, S. Cherkaoui, and Z. Mlika, "Federated learning and proactive computation reuse at the edge of smart homes," *IEEE Transactions on Network Science and Engineering*, 2021.

[3] D. Guliani, F. Beaufays, and G. Motta, "Training speech recognition models with federated learning: A quality/cost framework," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3080–3084.

[4] X. Zhu, J. Wang, Z. Hong, and J. Xiao, "Empirical studies of institutional federated learning for natural language processing," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 625–634.

[5] H.-Y. Chen and W.-L. Chao, "FedBE: Making bayesian model ensemble applicable to federated learning," in *Proc. ICLR*, 2021.

[6] T. Yang, G. Andrew, H. Eichner *et al.*, "Applied federated learning: Improving google keyboard query suggestions," *arXiv:1812.02903*, 2018.

[7] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," *arXiv:1906.04329*, 2019.

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

[9] R. S. Antunes, C. André da Costa, A. Küderle, I. A. Yari, and B. Eskofier, "Federated learning for healthcare: Systematic review and architecture proposal," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–23, 2022.

[10] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–37, 2022.

[11] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *Proc. Advances in Neural Information Processing Systems*, vol. 33, pp. 3557–3568, 2020.

[12] Y. Zhao *et al.*, "Federated learning with Non-IID data," *arXiv:1806.00582*, 2018.

[13] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Proc. NeurIPS*, 2020.

[14] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," in *Proc. ICLR*, 2021.

[15] M. Gao *et al.*, "Consistency-based semi-supervised active learning: Towards minimizing labeling cost," in *Proc. Springer ECCV*, 2020.

[16] Z. Lai *et al.*, "Joint semi-supervised and active learning for segmentation of gigapixel pathology images with cost-effective labeling," in *Proc. IEEE/CVF CVPR*, 2021.

[17] Y. Wang, S. Mukherjee, H. Chu, Y. Tu, M. Wu, J. Gao, and A. H. Awadallah, "Meta self-training for few-shot neural sequence labeling," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1737–1747.

[18] E. Bellocchio, F. Crocetti, G. Costante, M. L. Fravolini, and P. Valigi, "A novel vision-based weakly supervised framework for autonomous yield estimation in agricultural applications," *Engineering Applications of Artificial Intelligence*, vol. 109, p. 104615, 2022.

[19] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National science review*, vol. 5, no. 1, pp. 44–53, 2018.

[20] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated semi-supervised learning with inter-client consistency & disjoint learning," in *Proc. ICLR*, 2021.

[21] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, 2021.

[22] J. Pang, Y. Huang, Z. Xie, Q. Han, and Z. Cai, "Realizing the heterogeneity: a self-organized federated learning framework for iot," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3088–3098, 2020.

[23] R. Roelofs *et al.*, "A meta-analysis of overfitting in machine learning," *Proc. NeurIPS*, 2019.

[24] C. Shannon, "The zero error capacity of a noisy channel," *IRE Transactions on Information Theory*, vol. 2, no. 3, pp. 8–19, 1956.

[25] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *Proc. IEEE INFOCOM*, 2020.

[26] C.-W. Ching, Y.-C. Liu, C.-K. Yang, J.-J. Kuo, and F.-T. Su, "Optimal device selection for federated learning over mobile edge networks," in *Prof. IEEE ICDCS Workshop on NMIC*, 2020.

[27] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[28] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *Proc. ICLR*, 2017.

[29] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Proc. NeurIPS*, 2017.

[30] Y. Tang, W. Chen, Y. Luo, and Y. Zhang, "Humble teachers teach better students for semi-supervised object detection," in *Proc. IEEE/CVF CVPR*, 2021.

[31] J. Ling, L. Liao, M. Yang, and J. Shuai, "Semi-supervised few-shot learning via multi-factor clustering," in *Proc. IEEE/CVF CVPR*, 2022.

[32] W. Shi, Y. Gong, C. Ding, Z. M. Tao, and N. Zheng, "Transductive semi-supervised deep learning using min-max features," in *Proc. Springer ECCV*, 2018.

[33] V. S. Lokhande, S. Tasneeyapant, A. Venkatesh, S. N. Ravi, and V. Singh, "Generating accurate pseudo-labels in semi-supervised learning and avoiding overconfident predictions via hermite polynomial activations," in *Proc. IEEE/CVF CVPR*, 2020.

[34] B. Zhang *et al.*, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," *Proc. NeurIPS*, 2021.

[35] W. Zhang, L. Zhu, J. Hallinan, S. Zhang, A. Makmur, Q. Cai, and B. C. Ooi, "Boostmis: Boosting medical image semi-supervised learning with adaptive pseudo labeling and informative active annotation," in *Proc. IEEE/CVF CVPR*, 2022.

[36] Z.-H. Zhou, *Ensemble learning*. Springer, 2021.

[37] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 241–258, 2020.

[38] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–36, 2017.

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF CVPR*, 2018.

[40] F. Boccardi *et al.*, "Why to decouple the uplink and downlink in cellular networks and how to do it," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 110–117, 2016.

[41] C.-W. Ching, T.-C. Lin, K.-H. Chang, C.-C. Yao, and J.-J. Kuo, "Model partition defense against gan attacks on collaborative learning via mobile edge computing," in *Proc. IEEE GLOBECOM*, 2020.

[42] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Master's thesis, University of Toronto*, 2009.

[43] "Spam text message classification," 2017. [Online]. Available: https://www.kaggle.com/datasets/team-ai/spam-text-message-classification

[44] S. K. Lo *et al.*, "Architectural patterns for the design of federated learning systems," *Journal of Systems and Software*, p. 111357, 2022.

[45] W. Zhuang, X. Gan, Y. Wen, and S. Zhang, "Easyfl: A low-code federated learning platform for dummies," *IEEE Internet of Things Journal*, 2022.

[46] S. Chen, C. Shen, L. Zhang, and Y. Tang, "Dynamic aggregation for heterogeneous quantization in federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6804–6819, 2021.

[47] Z. Wang *et al.*, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *Proc. IEEE INFOCOM*, 2021.

[48] Y. Jin *et al.*, "Resource-efficient and convergence-preserving online participant selection in federated learning," in *Proc. IEEE ICDCS*, 2020.

[49] J. Zhang, N. Li, and M. Dedeoglu, "Federated learning over wireless networks: A band-limited coordinated descent approach," in *Proc. IEEE INFOCOM*, 2021.

[50] S. Wang *et al.*, "Device sampling for heterogeneous federated learning: Theory, Algorithms, and Implementation," in *Proc. IEEE INFOCOM*, 2021.

[51] L. Li *et al.*, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *Proc. IEEE INFOCOM*, 2021.

[52] X. Zhang, J. Wang, G. Joshi, and C. Joe-Wong, "Machine learning on volatile instances," in *Proc. IEEE INFOCOM*, 2020.

[53] N. H. Tran *et al.*, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE INFOCOM*, 2019.

[54] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *Proc. NeurIPS*, 2020.

[55] K. Hara, D. Saitoh, and H. Shouno, "Analysis of dropout learning regarded as ensemble learning," in *Proc. Springer ICANN*, 2016.

[56] N. Golrezaei, P. Mansourifard, A. F. Molisch, and A. G. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 7, pp. 3665–3676, 2014.

[57] G. Ding *et al.*, "Cellular-base-station-assisted device-to-device communications in tv white space," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 1, pp. 107–121, 2015.

[58] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[59] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.

[60] (2022) Bert question and answer. [Online]. Available: https://www.tensorflow.org/lite/examples/bert_qa/overview

[61] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805*, 2018.

[62] F. Kherif and A. Latypova, "Principal component analysis," in *Machine Learning*. Elsevier, 2020, pp. 209–225.

[63] S. Auer *et al.*, "Dbpedia: A nucleus for a web of open data," in *The semantic web*. Springer, 2007, pp. 722–735.

[64] Text classification with the torchtext library. [Online]. Available: https://pytorch.org/tutorials/beginner/text_sentiment_ngrams_tutorial.html

[65] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *Proc. NeurIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[66] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *Proc. MLSys*, 2019.

[67] P. Han, S. Wang, and K. K. Leung, "Adaptive gradient sparsification for efficient federated learning: An online learning approach," in *Proc. IEEE ICDCS*, 2020.

[68] T. Huang *et al.*, "Physical-layer arithmetic for federated learning in uplink mu-mimo enabled wireless networks," in *Proc. IEEE INFOCOM*, 2020.

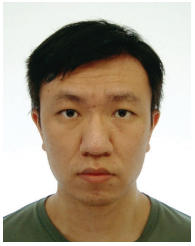
[69] Z. Zhang *et al.*, "Improving semi-supervised federated learning by reducing the gradient diversity of models," in *Proc. IEEE Big Data*, 2021.

[70] L. Che *et al.*, "FedTriNet: A pseudo labeling method with three players for federated semi-supervised learning," in *Proc. IEEE Big Data*, 2021.

[71] Y. Kang, Y. Liu, and X. Liang, "Fedcvt: Semi-supervised vertical federated learning with cross-view training," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 4, pp. 1–16, 2022.

[72] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Proc. NeurIPS*, 2017.

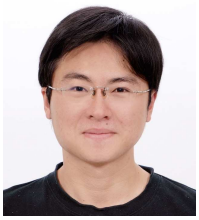
[73] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Proc. PMLR ICML*, 2021.



Cheng-Wei Ching (S'19-M'21) received the B.A. degree from the Department of Foreign Language, Tamkang University, Taiwan, in 2015. He received M.S. degree from the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan, in 2021. He is currently working toward the PhD degree in the Department of Computer Science and Engineering, UCSC, USA. His research interests include approximation algorithm, mobile edge computing, and machine learning.

Jia-Ming Chang received the B.S. and M.S. degrees in computer science from National Chiayi University, Taiwan, in 2020, and National Chung Cheng University, Taiwan, in 2022. He is currently with Taiwan Semiconductor Manufacturing Co., Ltd., as a Software Engineer. His research interests include reinforcement learning and federated learning.

Jian-Jhih Kuo (S'13-M'16) received the B.S. and Ph.D. degrees in computer science from National Chung Cheng University, Taiwan, in 2008, and National Tsing Hua University, Taiwan, in 2014. He was a Postdoctoral Fellow in the Institute of Information Science, Academia Sinica, Taiwan. He joined National Chung Cheng University, Taiwan, in 2018. He is currently an Associate Professor in the Department of Computer Science and Information Engineering. His research interests include mobile edge computing, cloud computing, software-defined networking, and quantum networking. He was a recipient of Ministry of Science and Technology Project for Excellent Junior Research Investigators in 2022 and National Chung Cheng University Young Faculty Award in 2021.



Chih-Yu Wang received the B.S. and Ph.D. degrees in electrical engineering and communication engineering from National Taiwan University (NTU), Taipei, Taiwan, in 2007 and 2013, respectively. He has been a visiting student in University of Maryland, College Park in 2011. He joined Academia Sinica, Taipei, Taiwan in 2014. He is currently an Associate Research Fellow / Associate Professor in Research Center for Information Technology Innovation. His research interests include game theory, wireless communications, social networks, and data science.

He was a recipient of the Ta-You Wu Memorial Award from National Science and Technology Council, Taiwan in 2022, the Young Scholars' Creativity Award from Foundation for the Advancement of Outstanding Scholarship in 2021, the Exploration Research Award 2021 from the Pan Wen-Yuan Foundation in 2021, the K. T. Li Young Researcher Award from ACM Taipei/Taiwan Chapter and The Institute of Information and Computing Machinery in 2019, Ministry of Science and Technology Research Project for Excellent Young Scholars in 2019 and 2022. His works were featured in 2018 and 2019 Significant Research Achievements of Academia Sinica. He is an IEEE Senior Member.