Federated Partially Supervised Learning with Limited Decentralized Medical Images

Nanqing Dong, *Student Member, IEEE*, Michael Kampffmeyer, *Member, IEEE*, Irina Voiculescu, *Member, IEEE* and Eric Xing, *Fellow, IEEE*

***Abstract*— Data government has played an instrumental role in securing the privacy-critical infrastructure in the medical domain and has led to an increased need of federated learning (FL). While decentralization can limit the effectiveness of standard supervised learning, the impact of decentralization on partially supervised learning remains unclear. Besides, due to data scarcity, each client may have access to only limited partially labeled data. As a remedy, this work formulates and discusses a new learning problem *federated partially supervised learning* (FPSL) for limited decentralized medical images with partial labels. We study the impact of decentralized partially labeled data on deep learning-based models via an exemplar of FPSL, namely, *federated partially supervised learning multi-label classification*. By dissecting FedAVG, a seminal FL framework, we formulate and analyze two major challenges of FPSL and propose a simple yet robust FPSL framework, `FedPSL`, which addresses these challenges. In particular, `FedPSL` contains two modules, *task-dependent model aggregation* and *task-agnostic decoupling learning*, where the first module addresses the weight assignment and the second module improves the generalization ability of the feature extractor. We provide a comprehensive empirical understanding of FSPL under data scarcity with simulated experiments. The empirical results not only indicate that FPSL is an under-explored problem with practical value but also show that the proposed `FedPSL` can achieve robust performance against baseline methods on data challenges such as data scarcity and domain shifts. The findings of this study also pose a new research direction towards label-efficient learning on medical images.**

***Index Terms*— Partially supervised learning, federated learning, multi-label classification**

I. Introduction

FUELED by the advances in deep learning research, *partially supervised learning* (PSL) [1]–[9] has emerged as a research direction for label-efficient learning on medical images, considering the practical issues such as data scarcity and high annotation cost. The problem of PSL, also known as the *missing annotations problem* [1] or *partial labels problem* [8] in the literature, is a family of learning tasks where the training data are partially labeled. Unlike commonly seen labeled data and unlabeled data, the definition of partially labeled data is associated with *multi-task learning* (MTL) [10]: given a task of interest that can be decomposed into multiple sub-tasks, an instance is only annotated for a subset of sub-tasks. In the medical domain, the problem of PSL commonly arises from the collection of multiple datasets from different sources for the task of interest, where each dataset is annotated for a specific sub-task as the annotation process usually requires relevant expertise. This makes all these datasets partially labeled when the task of interest includes all these sub-tasks.

As the datasets are acquired from different sources (*e.g.* different hospitals), the partially labeled datasets might be stored separately in different locations without direct connections. In the medical domain, data regulations commonly refer to situations that the data stored in the client are not allowed to be transferred to the server or other clients. A detailed explanation

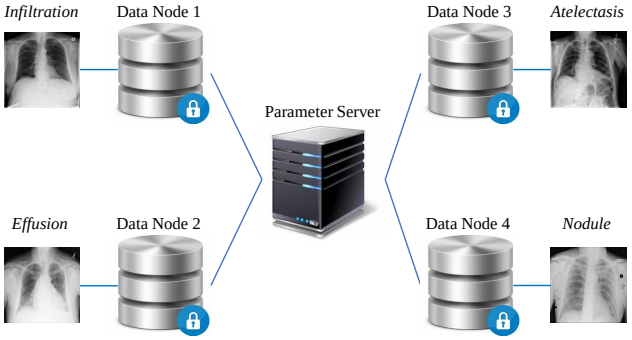

Fig. 1: Illustration of FPSL for a multi-label classification task on chest X-ray images. Here, each client (data node) is annotated for only one thoracic disease. We use this simple example to convey the main concept of the problem of interest (in practice, each client could be partially labeled for multiple classes). In this scenario, we only know whether each image in the first data node has *infiltration* but have no knowledge on the other three diseases. To ensure data government, only model weights and the metadata (*e.g.* statistics) of the local data can be communicated between each data node and the parameter server (see Sec. IV for a formal description). The goal of FPSL is to utilize the four partially labeled datasets stored in the different data nodes to train the model of interest in the parameter server.

This work was partially funded by the Research Council of Norway grants no. 315029, 309439, and 303514.

N. Dong and I. Voiculescu are with the Department of Computer Science, University of Oxford, Oxford, OX1 3QD, UK. (email: nanqing.dong@cs.ox.ac.uk, irina.voiculescu@cs.ox.ac.uk)

M. Kampffmeyer is with the Department of Physics and Technology at the University of Tromsø, 9019 Tromsø, Norway. (email: michael.c.kampffmeyer@uit.no)

E. Xing is with the Machine Learning Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA; and also with Mohamed bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi, UAE (email: epxing@cs.cmu.edu)

is given in Sec. IV. These regulations might be made and supervised by either the data holder or even the government (*e.g.* EU General Data Protection Regulation [11] and US Health Insurance Portability and Accountability Act [12]). Thus, it is natural to think about the connection between PSL and *federated learning* (FL) [13]. FL is a learning paradigm that aims to utilize decentralized data stored separately in different places and has become a topic of active research in medical image analysis [14]–[19].

In this work, we extend the problem formulation of PSL to a federated setup and formulate *federated partially supervised learning* (FPSL) for medical images. As one of the core contributions of this work, a formal problem definition of FPSL is provided in Sec. IV. To the best of our knowledge, this is the first study of FPSL. For an intuitive understanding, a concrete example of FPSL is illustrated in Fig. 1. It is worth noting that a direct combination of FL and PSL does not provide a robust solution to the problem of interest. Firstly, the federated setup poses a non-trivial barrier for the implementation of some PSL methods. For example, VRM-based PSL methods [8], [9] require access to the training data in a centralized fashion. Another popular family of PSL methods, label propagation-based PSL methods [3] involve iterated training over each partially labeled dataset to generate pseudo labels. In contrast to semi-supervised learning, where labels are instance-wise (each instance is either labeled or unlabeled), PSL has task-wise labels (each task is either labeled or unlabeled for a given instance). Thus, in PSL, more iterations should be consumed to ensure the quality of pseudo labels for certain tasks, which leads to a low efficiency in computation under a federated environment. Secondly, in the medical domain, the partially labeled datasets stored in the clients (data nodes) are commonly small, *i.e.* the partial labels are scarce, which means the local data might not be able to support the efficient training of a model with complex network architecture [4], [7]. Thirdly, in FL, the data are assumed to be non-independent and identically distributed (non-IID). Generally, non-IID describes the situation that each client (*e.g.* hospitals) collect data from different populations. In this work, as the data are partially labeled, clients can have different label distributions. But, none of the existing PSL methods have tried to tackle these challenges.

Before presenting our solution, we first examine an existing FL framework, FedAVG [13], which has served as a seminal baseline in FL for fully labeled data. By analyzing FedAVG under FPSL, we suggest that there are two major challenges of FPSL. Firstly, in the model aggregation step, the aggregation weights should reflect the impact of partial supervision. Secondly, a direct consequence of label scarcity and class imbalance in the local training phase is overfitting. Especially, when each client only has partial labels with respect to a few classes, the features extracted by the learned model might not be able to generalize well to unseen classes. It is important to mitigate the local overfitting by learning robust features. To address the two aforementioned challenges, we present a simple yet robust FPSL framework `FedPSL` based on FedAVG [13]. `FedPSL` consists of two modules that are designed to address the two challenges respectively. The first module is a *task-dependent model aggregation* (TDMA) module. The model (a neural network) is decomposed into two parts: a feature extractor and a predictor. The aggregation of the predictor in the *parameter server* [20] is dependent on the sub-tasks across the clients (*i.e.* each class is considered as a sub-task under the perspective of MTL). The second module is a *task-agnostic decoupling learning* (TADL) module. The local feature extractor could suffer from overfitting caused by both partial supervision and data scarcity. We aim to improve the generalization ability of the local feature extractor by decoupling the learning process of the feature extractor and predictor. To achieve this goal, we first reformulate the optimization objective as a *bi-level optimization* problem [21], where the feature extractor and the predictor are optimized on different data splits. Then, by using *meta-learning* [22], we perform meta-optimization on the feature extractor to alleviate the overfitting.

As the first study in FPSL, a primary goal in this work is to provide a comprehensive empirical understanding on FPSL under data scarcity. In addition, we aim to evidence the contributions of `FedPSL`. Specifically, we aim to provide an empirical understanding on the effects of data scarcity and class imbalance under the federated setup, which both can lead to overfitting. Here, the term "data scarcity", also known as "label scarcity" refers to the situation that only limited partial labels are available in the clients. The term "class imbalance" has two meanings: i) the classes with more partial labels can dominate the learning process, and ii) for each class, there are more negative examples than positive examples. Without loss of generality, we illustrate FPSL with multi-label classification (MLC), a representative task prone to overfitting in a federated setup. MLC is a fundamental yet challenging task as it does not have mutually exclusive classes. In contrast to multi-class settings, we can not utilize the constraint of mutually exclusive classes as prior knowledge in either loss formulation [4], [5] or data augmentation [8], which are utilized in centralized PSL. We evaluate `FedPSL` against strong baselines in terms of both performance and robustness under various data challenges such as label scarcity and class imbalance. The empirical results show that `FedPSL` can consistently outperform the baseline methods and can be used as a robust framework for FPSL.

The contributions can be summarized as follows:

1) We formulate and discuss for the first time the problem of FPSL for decentralized medical images, and propose `FedPSL`, a simple and robust framework for federated partially supervised multi-label classification under data scarcity.
2) We formulate and explain the challenges of FPSL.
3) We propose a novel federated partially supervised training pipeline including a task-dependent model aggregation module and a task-agnostic decoupling learning module.
4) We show initial evidence that FPSL is an under-explored problem compared with existing learning paradigms and offer the community the first benchmark of federated partially supervised multi-label classification, accompanied with a set of performance evaluations and baseline

comparisons.

The rest of this paper is organized as follows. Sec. II reviews the relevant literature for FL and PSL and Sec. IV formally formulates FPSL, the problem of interest. Sec. V-C theoretically analyzes the challenges of FPSL. Sec. VI describes the proposed solution in detail. Sec. VII describes the proposed benchmark tasks and provides experimental results and analysis. Section VIII summarizes this work.

II. Related Work

A. Federated Learning

There are few FL studies directly related to FPSL. Three related areas are federated unsupervised representation learning (FURL) [23], federated *positive-unlabeled* (PU) learning [24], and federated semi-supervised learning (*semi*-SL) [25]. FedU [23] present a divergence-aware update mechanism for FURL. However, FedU only considers the local updates rather than global aggregation. As a federated extension of PU learning, FedAwS [24] shares a similar problem formulation as FPSL by assuming that each client only has access to labels of one class. However, FedAwS is designed for multi-class classification only. That is to say, each client will have both fully labeled and unlabeled data, and thus differs from the partial labels problem discussed in this work. As *semi*-SL has been successfully applied to PSL, federated *semi*-SL is another related domain to FPSL. The state-of-the-art federated *semi*-SL method FedMatch [25], for instance, adopts a pseudo-labeling training strategy based on consistency regularization. However, FPSL methods that are based on *semi*-SL exhibit large variety in terms of class-wise performance, as the quality of pseudo labels are dependent on the amount of available partial labels. We will illustrate this point via experiments in Sec. VII-E.

B. Partially Supervised Learning

Recently, there have been efforts made to utilize multiple partially labeled datasets in the medical domain. However, none of these methods are designed for the situation that the partially labeled datasets are decentralized. [8], [9] address the partial labels issue by generating vicinal labels based on human structure similarity, which can only be implemented in a centralized training environment. [3], [5] both require a fully labeled dataset in the training process. It is less practical to assume that fully labeled data are available in each client, and only having one client or a few clients with fully labeled data will inevitably impair the learning process in contrast to centralized training. Besides, PSL methods that are based on label propagation [3] have iterating training procedures, which not only increase the complexity of a federated implementation but also lead to sub-optimal performance. A practical issue that is often ignored in the medical domain is that there are only limited labeled data available. In this work, we denote the situation that only limited partial labels are available in each client as *label scarcity*. PSL methods with complex network architectures or training procedures [4], [7] normally perform much worse than counterparts that have access to large-scale training data in a centralized environment [8]. With small-scale local training data in each client, the issue becomes more challenging in a federated environment. As existing PSL methods struggle in the problem formulation defined in Sec. IV), it is important to study the problem of FPSL and develop a robust solution.

III. Preliminaries

1) Partially Supervised Learning: Before we formulate the problem of federated partially supervised learning, we briefly review *partially supervised learning* (PSL). Given a task of interest, suppose there are $C > 1$ classes of interest indexed by the set $\mathcal{C}$. Let x denote an image instance, y_{full} denote the corresponding complete label of x with the label set $\mathbb{S}(y_{\text{full}}) \subset \mathcal{C}$, where $\mathbb{S}(\cdot)$ is a set operation.[1] Analogous to y_{full}, we define the incomplete label or *partial label* of x as y_{part} with the label set $\mathbb{S}(y_{\text{part}})$. Here, we require $|\mathbb{S}(y_{\text{part}})| \neq \emptyset$ and $|\mathbb{S}(y_{\text{part}})| \subset |\mathbb{S}(y_{\text{full}})|$, *i.e.* $0 < |\mathbb{S}(y_{\text{part}})| < |\mathbb{S}(y_{\text{full}})|$, where $|\cdot|$ is the cardinality. For simplicity, we use y to denote the partial label with respect to x in the remainder of the paper.

Without loss of generality, we assume that the partially labeled dataset $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{S}|}$ can be split into K sub-datasets where each sub-dataset contains label information of a few classes, *i.e.* $\mathcal{S} = \bigcup_k^K \mathcal{S}_k$. Here, $\mathcal{S}_k = \{(x_i^k, y_i^k)\}_{i=1}^{|\mathcal{S}_k|}$ denotes the partially labeled dataset in the k^{th} sub-dataset and y_i^k is the partial label of the example x_i^k with $\mathbb{S}(y_i^k) = \mathcal{C}_k \subset \mathcal{C}$ where $\mathcal{C}_k$ is the class set for the k^{th} sub-dataset. For a better illustration of PSL, a common task is presented below as a concrete example.

2) Multi-Label Classification: As a generalization of multi-class classification, a multi-label classification (MLC) task could be interpreted as C binary classification tasks. In contrast to multi-class classification, the classes in MLC are not mutually exclusive, *i.e.* each image instance could belong to more than one category at the same time. For example, a chest X-ray image could be diagnosed as cardiomegaly and emphysema simultaneously. Mathematically, given the input image space $\mathcal{X}$, $\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R}^C\}$ is a family of functions of interest. For *partially supervised multi-label classification* [9], each sub-dataset is only annotated for a true subset of $\mathcal{C}$.

IV. Problem Setup

Now, we formulate the problem of *federated partially supervised multi-label classification*, an exemplar of FPSL. Analogous to Sec. III-.1, we have K clients (data nodes) in a federated system and $\mathcal{S}_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$ denotes the partially labeled dataset stored in the k^{th} client. Following standard practice in FL, we assume $\mathcal{S}_k \cap \mathcal{S}_l = \emptyset$ for $k \neq l$ and $\{\mathcal{S}_k\}_{k=1}^K$ are all non-IID data. In addition to the partially labeled data, unlabeled datasets $\{\mathcal{U}_k\}_{k=1}^K$ might also be available in each client. Given a model of interest f_θ and an independent fully labeled target dataset $\mathcal{T} = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ that is unseen during the training, the learning outcome is to find the optimal parameter set θ that minimizes the estimated *empirical risk*:

$$\hat{\mathcal{R}}_\theta = \frac{1}{n_t} \sum_i^{n_t} \mathcal{L}(f_\theta(x_i^t), y_i^t) = \mathcal{L}_{\mathcal{T}}(f_\theta), \tag{1}$$

[1]Here, we use $\subset$ instead of $=$ because x might not contain all classes.

where $L(\cdot,\cdot)$ is the loss function.

In this work, we consider a seminal FL setup, where each data node (client) is only connected to a master node (server). The master node does not store any clinical data and could be implemented as a *parameter server* (PS) [20]. In addition to the standard setup of FL, the privacy-critical nature of the medical domain imposes another constraint: the transferring of clinical data between the master node and data nodes is prohibited. That is to say, only model weights and *metadata* (*e.g.* statistics of data) [26] should be communicated across nodes. In contrast to the common data privacy issues in FL, the data holders are prevented from exchanging user data in any form to ensure data government. For example, a hospital might not be allowed to upload the patients' data stored in its server to another institute. With this new constraint, FPSL on medical images is more challenging than a simple integration of FL and PSL.

It is worth mentioning that the primary goal of this work is to formulate FPSL on medical images to facilitate data government. To obtain privacy-preserving guarantees, an integration of differential privacy [27] techniques such as DPSGD [28] will be required, which, while being out of the scope of this work, is a promising direction for future work.

V. Towards Understanding FPSL

FPSL is an emerging yet practical problem in the medical domain. Yet, there is limited analysis in the literature. In this section, we aim to provide the first preliminary theoretical understanding of FPSL, which also motivates our method in Sec. VI. We continue to use federated partially supervised multi-label classification as an example.

A. A Multi-Task Representation of MLC

To ease the analysis below, we first describe the mathematical formulation of MLC. A multi-label classifier can be denoted as $g_\phi \circ f_\theta$, where f_θ is the feature extractor and g_ϕ is the multi-label predictor. Given an input x, we have $(g_\phi \circ f_\theta)(x) = g_\phi(f_\theta(x))$.

If the multi-label predictor is a fully-connected layer [29], the parameters of the multi-label predictor ϕ can be further represented as a $C \times d$ matrix, where d is the dimension of the output feature vector $f_\theta(x) \in \mathbb{R}^d$ extracted by the feature extractor. In this way, we can represent the classification network output as

$$(g_\phi \circ f_\theta)(x) = \phi \cdot f_\theta(x), \tag{2}$$

where $\cdot$ denotes the dot product operation. The optimization goal is then

$$\min_{\theta,\phi} \mathcal{L}_\mathcal{S}(g_\phi \circ f_\theta). \tag{3}$$

More specifically, given C classes of interest, we can decompose the weight matrix ϕ into C weight vectors, *i.e.*

$$\phi = \begin{bmatrix} \phi^1 \\ \phi^2 \\ \vdots \\ \phi^C \end{bmatrix}. \tag{4}$$

For a single class c, the probability score of the prediction is then

$$p_c(x) = \text{sigmoid}(\phi^c \cdot f_\theta(x)), \tag{5}$$

where $\text{sigmoid}(z) = \frac{1}{1+\exp^{-z}}$.

B. A Closer Look at FedAVG

Given the formulation of MLC, to understand the difference between FPSL and standard supervised FL, we dissect a seminal FL method, FedAVG [13]. Given $\{\mathcal{S}_k\}_{k=1}^K$ stored in K clients, there are K corresponding multi-label classifiers, denoted as $\{g_{\phi_k} \circ f_{\theta_k}\}_{k=1}^K$. For FedAvg [13], $\{\theta_k\}_{k=1}^K$ and $\{\phi_k\}_{k=1}^K$ are aggregated into θ_0 and ϕ_0 in the PS by

$$\begin{aligned} \theta_0 &= \sum_k w_k \theta_k, \\ \phi_0 &= \sum_k w_k \phi_k, \end{aligned} \tag{6}$$

where $w_k = \frac{n_k}{\sum_k n_k}$ and $\sum_k w_k = 1$. As we assume supervised FL here, n_k is just the number of labeled instances in each client k.

Without loss of generality, let us look at a single class c. Following the formulation of Eq. (2) and Eq. (6), the final prediction model can be written as

$$\begin{aligned} g_{\phi_0^c} \circ f_{\theta_0}(x) &= \phi_0^c \cdot f_{\theta_0}(x) \\ &= \sum_k w_k \phi_k^c \cdot \sum_k w_k f_{\theta_k}(x) \\ &= \sum_{i,j} w_i w_j \phi_i^c \cdot f_{\theta_j}(x). \end{aligned} \tag{7}$$

Eq. (7) is fairly robust under various setups when $\{\theta_k\}_{k=1}^K$ and $\{\phi_k^c\}_{k=1}^K$ are *reliable*. In a hypothetical scenario, if $\{\theta_k\}_{k=1}^K$ and $\{\phi_k^c\}_{k=1}^K$ both share the same properties of the *oracle* functions θ^* and ϕ^{c*}, Eq. (7) implies $g_{\phi_0^c} \circ f_{\theta_0}(x) = g_{\phi^{c*}} \circ f_{\theta^*}(x)$.

C. Challenges of FPSL

The hypothetical scenario in Sec. V-B, where each client has fully labeled data, is however, rarely the case in practice, especially in the medical domain. Under FPSL, the labels can be partial at each client, which means $\{\theta_k\}_{k=1}^K$ and $\{\phi_k^c\}_{k=1}^K$ might not be as *reliable* as assumed in standard supervised FL. In addition to partial supervision, each client might not have enough labels for each class to learn meaningful representations.

Again, let us look at a single class c. Assume that $K > 1$ clients are split into two sets, which are $\mathcal{K}_L$ and $\mathcal{K}_U$. $\mathcal{K}_L$ denotes the clients that contain partial labels with respect to class c and $\mathcal{K}_U$ denotes the clients that do not contain any partial labels with respect to class c. Under this definition,

Eq. (7) can be reformulated as

$$
\begin{aligned}
g_{\phi_0^c} \circ f_{\theta_0}(x) = & \sum_{i_L \in \mathcal{K}_L, j_L \in \mathcal{K}_L} w_{i_L} w_{j_L} \phi_{i_L}^c \cdot f_{\theta_{j_L}}(x) \\
& + \sum_{i_L \in \mathcal{K}_L, j_U \in \mathcal{K}_U} w_{i_L} w_{j_U} \phi_{i_L}^c \cdot f_{\theta_{j_U}}(x) \\
& + \sum_{i_U \in \mathcal{K}_U, j_L \in \mathcal{K}_L} w_{i_U} w_{j_L} \phi_{i_U}^c \cdot f_{\theta_{j_L}}(x) \\
& + \sum_{i_U \in \mathcal{K}_U, j_U \in \mathcal{K}_U} w_{i_U} w_{j_U} \phi_{i_U}^c \cdot f_{\theta_{j_U}}(x).
\end{aligned} \tag{8}
$$

1) Impact on Task-Specific Predictor: Let us assume that f_{θ_k} in Eq. (8) can achieve the same performance of the *oracle* function and focus instead on the task-specific predictor g_{ϕ_k}. For unlabeled clients $\mathcal{K}_U$ (with respect to class c), no contributions should be made to the model aggregation as no label information are utilized in the local training. Thus, the third and the fourth terms of Eq. (8) will inevitably degrade the final performance. At the beginning of the federated training, $\{\phi_k\}_{k \in \mathcal{K}_U}$ are randomly initialized or initialized with unsupervised pre-trained weights. During the local training, $\{\phi_k\}_{k \in \mathcal{K}_U}$ are untouched as no label information are involved in the training of the task-specific predictors. Thus, $\{\phi_k\}_{k \in \mathcal{K}_U}$ are only updated when synchronized with the global model weights, *i.e.* they have to wait to be updated by propagating the error of initialization. Compared with $\{\phi_k\}_{k \in \mathcal{K}_L}$, which are optimized in each local training round, $\{\phi_k\}_{k \in \mathcal{K}_U}$ are de facto *stragglers* [30] in distributed ML. This means that $\{\phi_k\}_{k \in \mathcal{K}_U}$ not only lower the performance but also slow down the convergence.

One might argue that, with large-scale training, FedAVG or advanced variants of FedAVG (*e.g.* FedProx [31]) can automatically mitigate this compound negative effect caused by decentralization and partial supervision. However, it is less practical to collect such large-scale annotations in the medical domain. On the contrary, under the problem formulation of this work, the limited partial labels will further exacerbate the situation.

Moreover, let us consider an extreme case. Say class c is a rare disease and the unlabeled datasets in clients $k \in \mathcal{K}_U$ are relatively larger in size than the partially labeled ones in clients $k \in \mathcal{K}_L$, *i.e.* $w_i \gg w_j \ \forall i \in \mathcal{K}_U, j \in \mathcal{K}_L$. This will make the fourth term a dominating term in (8), which can significantly deteriorate the FL system, for the same reasons mentioned above.

2) Impact on Task-Agnostic Feature Extractor: Now, let us relax the assumption in Sec. V-C.1 and analyze the behavior of f_θ. In standard SL, θ and ϕ are optimized together by minimizing Eq. (1). Thus, under a more *Bayesian* view, the prediction can also be represented as $(g_{\phi^c|\theta} \circ f_\theta)(x)$, *i.e.* ϕ^c is dependent on θ (or in another direction, θ is dependent on ϕ^c). With large-scale fully labeled data, such a dependence is normally ignored, as in each client, f_θ should extract information for all classes of interest more or less. However, under FPSL, the clients $\mathcal{K}_U$ are not optimized to capture features with respect to c. This can cause non-trivial *weight divergence* [32] across the clients $\mathcal{K}_U$ and the clients $\mathcal{K}_L$ in the system.

Eq. (6) can also be interpreted as *ensemble learning* [33]. Ensemble learning aims to improve the model generalization ability by leveraging multiple classifiers [34]. However, under FPSL, due to partial supervision, the quality of f_θ in a client could be biased to a few classes. While this may have a positive effect on the first term of Eq. (8) for class c, the other classes can be negatively influenced by the last three terms. Again, with large-scale labeled data, this issue can be mitigated. However, with limited data, the overfitting could be severe. Ideally, θ should be task-agnostic to improve the generalization ability of the feature extractor.

VI. Method

Motivated by the limitations of FedAVG in Sec. V-C, we present `FedPSL`, a FPSL framework consisting of two simple yet efficient techniques that aim to better handle the weight assignment and efficiently utilize the limited partial supervision. First, we propose a model aggregation by considering the number of partial labels in Sec. VI-A. Second, we propose to decouple the feature extractor and predictor in the learning process in Sec. VI-B. The empirical advantages of `FedPSL` will be illustrated in Sec. VII.

A. Task-Dependent Model Aggregation

As discussed in Sec. V-C.1, the model aggregation on the predictor should be linked with class-wise partial labels across the clients, instead of the amount of data. To mitigate the negative impact of *wrong* weight assignments in FedAVG, the model aggregation is conducted separately for the feature extractor and the multi-label predictor. For the feature extractor f_θ, we adopt the same aggregation mechanism as Eq. (6). While, for the multi-label predictor g_ϕ, we aggregate the task-specific weights ϕ^c independently. We have

$$
\begin{aligned}
\theta_0 &= \sum_k \frac{n_k}{\sum_k n_k} \theta_k, \\
\phi_0^c &= \sum_k \frac{n_k^c}{\sum_k n_k^c} \phi_k^c,
\end{aligned} \tag{9}
$$

where n_k^c denotes the number of labeled examples in client k with respect to class c.

In contrast to Eq. (6), which is designed for standard SL, we use the number of partially labeled examples with respect to class c in each client to indicate the contribution of this client. For clients that do not contain label information of c, the contributions to the model aggregation will just be zeros. However, we still use n_k, instead of $\sum_c n_k^c$, to indicate the contribution to θ. This is because $\sum_c n_k^c \geq n_k$, which could exacerbate the negative effect of class imbalance on the feature extractor and the proposed solution aims to alleviate it. For the same reason, the feature extractor and the multi-label predictor should be decoupled in the local training to improve the generalization ability.

B. Task-Agnostic Decoupling Learning

As discussed in Sec. V-C.2, the feature extractor f_θ and the multi-label predictor g_ϕ are mutually dependent. This means

that the training of f_θ could get biased to the partially labeled classes in each client. To mitigate the local overfitting and improve the generalization ability of f_θ, we propose a local training strategy based on *meta-learning* [22].

Before presenting the training strategy, we first rephrase the local training target as a *bi-level optimization* [21], [35] problem. Assume the local training data $\mathcal{S}_k$ can be split into two subsets, one is denoted as the training set and the other one is denoted as the validation set. We propose the following formulation:

$$\begin{aligned} &\min_{\theta} \mathcal{L}_{val}(g_{\phi^*|\theta} \circ f_\theta(x), y); \\ &\text{s.t.} \qquad \phi^*|\theta = \arg\min_{\phi} \ \mathcal{L}_{train}(g_\phi \circ f_\theta(x), y), \end{aligned} \tag{10}$$

where θ is the upper level variable and ϕ is the lower level variable. Intuitively, we misalign the sample spaces of f_θ and g_ϕ to mitigate local overfitting.

It is worth mentioning that the formulation Eq. (10) only makes sense under data scarcity. Given large-scale fully labeled data, Eq. (10) actually leads to less inefficient training than standard joint supervised training, as θ and ϕ are optimized in different feature spaces. However, with only limited partially labeled data, we leverage Eq. (10) as a form of meta-learning to improve the generalization ability of f_θ.

A direct implementation of Eq. (10) is difficult and suffers from inefficient training. Thus, we approximate Eq. (10) with *stochastic gradient descent* (SGD). The idea is to approximate ϕ^* by adapting θ using only a single training step, without solving the inner optimization completely. The local training scheme in each client is illustrated in Algorithm 1. Note, in Line $6-7$, we evaluate the gradients with respect to both θ_k and ϕ_k with $\mathcal{B}_{train}$, instead of ϕ_k alone in Eq. (10). Line 9 depicts the meta-optimization step, where the meta-objective is

$$\begin{aligned} &\min_\theta \mathcal{L}_{\mathcal{B}_{val}}(g_{\phi_k} \circ f_{\theta_k^{'}}) = \\ &\qquad \min_\theta \mathcal{L}_{\mathcal{B}_{val}}(g_{\phi_k} \circ f_{\theta_k - \alpha \nabla_{\theta_k} \mathcal{L}_{\mathcal{B}_{train}}(g_{\phi_k} \circ f_{\theta_k})}) \end{aligned} \tag{11}$$

In our implementation, to efficiently utilize the available data, we do not split $\mathcal{S}_k$ into two non-overlapping subsets. Instead, when sampling a random mini-batch, we simply split the mini-batch into half to generate $\mathcal{B}_{train}$ and $\mathcal{B}_{val}$.

The complete federated training scheme is presented in Algorithm 2.

Algorithm 1 Local training procedure for client k.

$\mathcal{S}_k$: Partially labeled data in client k
E: Number of epochs
B: Number of batches
α, β: Learning rates
1: **function** CLIENT_UPDATE(θ_k, ϕ_k)
2: **for** $t = 1, 2, \cdots, E$ **do**
3: **for** $b = 1, 2, \cdots, B$ **do**
4: Sample $\mathcal{B}_{train}, \mathcal{B}_{val}$ from $\mathcal{S}_k$
5: *// Update ϕ_k with $\mathcal{B}_{train}$*
6: $\theta_k^{'} \leftarrow \theta_k - \alpha \nabla_{\theta_k} \mathcal{L}_{\mathcal{B}_{train}}(g_{\phi_k} \circ f_{\theta_k})$
7: $\phi_k \leftarrow \phi_k - \alpha \nabla_{\phi_k} \mathcal{L}_{\mathcal{B}_{train}}(g_{\phi_k} \circ f_{\theta_k})$
8: *// Meta-update θ_k with $\mathcal{B}_{val}$*
9: $\theta_k \leftarrow \theta_k - \beta \nabla_{\theta_k} \mathcal{L}_{\mathcal{B}_{val}}(g_{\phi} \circ f_{\theta_k^{'}})$
10: **return** θ_k, ϕ_k

Algorithm 2 Training procedure for `FedPSL`.

M: Metadata
T: Number of training rounds
1: **for** $t = 1, 2, \cdots, T$ **do**
2: **for** $k = 1, 2, \cdots, K$ **do**
3: $\theta_k, \phi_k \leftarrow \theta_0, \phi_0$ ▷ Synchronize with PS
4: $\theta_k, \phi_k \leftarrow$ Client_Update(θ_k, ϕ_k)
5: Upload $\{\theta_k\}_{k=1}^K$ and $\{\phi_k\}_{k=1}^K$ to PS
6: $\theta_0, \phi_0 \leftarrow$ Aggregate($\{\theta_k\}_{k=1}^K, \{\phi_k\}_{k=1}^K, M$) ▷ Eq. (9)

VII. Experiments

The purposes of the conducted experiments are threefold. Firstly, we aim to illustrate that FPSL is an under-explored yet challenging problem compared with standard FL and centralized training. Secondly, we want to discuss several initial solutions to FPSL. Thirdly, we want to demonstrate the robustness of `FedPSL` against label scarcity and class imbalance. We use a multi-label classification (MLC) task on chest X-ray images (CXRs) to evaluate the proposed framework. The labels for a MLC task are usually sparse (*e.g.* 60% of CXRs in ChestX-ray14 dataset [36] have no findings of thoracic diseases), which makes federated partially supervised MLC even more difficult.

A. Datasets

We utilize three public datasets to simulate the multi-site partially labeled datasets. The visual difference on the multi-site CXRs is illustrated in Fig. 2.

1) ChestX-ray14 Dataset: ChestX-ray14 dataset[2] [36] is a public CXR dataset for multi-label chest disease detection. It contains label information for 14 classes. Each CXR can contain multiple diseases at the same time.

2) Tuberculosis Chest X-ray Database: Tuberculosis Chest X-ray Database (Tuberculosis dataset)[3] [37] is a public CXR dataset, where each CXR is only annotated for tuberculosis.

3) COVID-19 Detection Dataset: COVID-19 Detection Dataset (COVID-19 dataset)[4] [38] is a public CXR dataset, where each CXR is annotated for COVID-19.

B. Experimental Setup

There are K clients in our experiments. The goal is to leverage K partially labeled datasets ($\mathcal{S}_1$ - $\mathcal{S}_K$) stored in K separated clients to learn a multi-label image classifier for C diseases. Note, to ensure data government, only model weights and metadata are allowed to be exchanged between the PS and each data node.

We follow [29], [36] and choose *area under receiver operating characteristic* (AUROC) as the evaluation metric in this

[2]https://nihcc.app.box.com/v/ChestXray-NIHCC/
[3]https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-dataset
[4]https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database

(a) ChestX-ray14 (b) Tuberculosis (c) COVID-19

Fig. 2: Visual illustration of multi-site CXRs. (a) is diagnosed with cardiomegaly and emphysema. (b) is tuberculosis positive. (c) is COVID-19 positive.

work. Note, AUROC does not specify the threshold, unlike precision, recall, or F1-score, and is thus preferred in our quantitative comparison. We report the mean over three runs with different random seeds. We select the best performance in each run based on the highest average AUROC of the C diseases.

C. Baselines

The choice of baseline methods gives consideration to two aspects. First, we want to provide an empirical understanding of FPSL. Second, we want to examine the performance of the seminal methods from existing learning paradigms such as SL, *semi*-SL, and *self*-SL, when addressing FPSL.

We first compare `FedPSL` against five robust FPSL baselines.

- `FedAVG` is an adaptation of FedAVG [13] to FPSL. To differentiate from FedAVG, we use `FedAVG` to denote the method. FedAVG is a seminal method that has robust performance in FL. In the data nodes, we only update the weights given the partial labels in the backpropagation. Note, this is equivalent to standard SL. In the PS, only the shared weights are aggregated and synchronized.
- `FedProx` is an adaptation of FedProx[5] [31] to FPSL. To differentiate from FedProx, which is designed for non-IID fully labeled data, we use `FedProx` to denote the method. We follow the same setup of `FedAVG` and use 0.001 for the proximal term.
- `FedSSP` denotes a learning paradigm of self-supervised pre-training followed by fine-tuning on partial labels. We adapt SimSiam[6] [39] to a federated environment. The prediction heads are fine-tuned based on partial labels, in a fashion similar to `FedAVG`.
- `FedMatch` is an adaptation of FedMatch[7] [25], a state-of-the-art federated *semi*-SL baseline based on consistency regularization training. As there is no existing PSL method designed for the problem of interest yet. We adapt FedMatch as a strong PSL baseline with default hyperparameters.
- `FedMatch+` follows the same setup of `FedMatch`, except that the feature extractor is pre-trained in the same way as `FedSSP`.

[5]https://github.com/litian96/FedProx
[6]https://github.com/facebookresearch/simsiam
[7]https://github.com/wyjeong/FedMatch

The second group includes four centralized baselines, where the corresponding constraints in Sec. IV are relaxed (*i.e.* centralized training is feasible or full labels are available). We include three centralized PSL methods to provide an empirical understanding of the impact of FPSL and a supervised *Oracle* with full labels.

- IML [2] is the simplest PSL method, which simply ignores missing labels, *i.e.* only backpropagating the gradients corresponding to the partial labels.
- FixMatch[8] is a centralized *semi*-SL method, which is adapted to PSL for the same reason of `FedMatch`. We use the same set of hyperparameters of `FedMatch`. This is also the centralized training counterpart of `FedMatch`.
- MixUp-PME [9] is a centralized PSL method based on data augmentation and pseudo labeling. We use the default hyperparameters of [9].
- *Oracle* is the standard SL baseline, which has centralized data with full labels. As expected, this is the best performance that the model of interest can achieve under standard SL, which is also considered as the upper bound performance for centralized fully labeled data.

D. Implementation

1) Data Pre-Processing: Each CXR has a resolution of 1024×1024. During both training and testing of the model, each CXR is first resized to 224×224 and the image is then normalized by instance normalization: $\hat{x}^{ij} = \frac{x^{ij}-\mu(x)}{\sigma(x)}$, where x is an image, $\hat{x}$ is the normalized image, (i, j) is the position of the pixel in a 224×224 image, and μ and σ are the mean and standard deviation of the pixels of x. We use the same data augmentation policy proposed in [26] for all methods in the training process.

2) Network Architecture: All baseline methods use a DenseNet121 [40] as the encoder f_θ. We choose DenseNet121 following [29] and because it is a commonly adopted model in FL [13] for a lightweight experimental setup.[9] Each of the federated methods has $K + 1$ DenseNet121s for K clients and the PS, while each of the non-federated methods has one DenseNet121. All models are implemented in PyTorch on an NVIDIA Tesla V100.

3) Training: For a fair comparison, all networks are initialized with the same random seeds. We train all methods with partial or full labels for 100 epochs. The synchronization and aggregation for the federated methods are performed every 10 epochs. We use a standard Adam [41] optimizer with a fixed learning rate $\alpha = \beta = 10^{-3}$ for supervised training or partially supervised training, *i.e.* we use the same learning rates for the standard SL setup and the meta-learning setup. The binary cross-entropy for each class is weighted by $\frac{N_{neg}}{N_{pos}}$, where N_{neg} and N_{pos} are the numbers of negative cases and positive cases for the class of interest in the labeled data. We use the PyTorch bi-level optimization package `higher`[10] to perform the meta-optimization step.

[8]https://github.com/google-research/fixmatch
[9]The experiments with more advanced architectures are considered out of the scope of this work.
[10]https://github.com/facebookresearch/higher

TABLE I: Label statistics of positive cases for the experimental setup **without** one-class clients in the federated system. "0" denotes that the client does not have any partial labels for the disease, *i.e.* this disease is unlabeled.

Disease	Clients			
	1	2	3	4
Infiltration	141	0	0	0
Effusion	131	98	0	0
Atelectasis	78	91	121	0
Nodule	0	45	67	55
Consolidation	0	0	40	49
Pneumothorax	0	0	0	45

E. Federated System without One-Class Clients

1) Data Setup: In the first experiment, we consider a common scenario, where each client has $1 < C_k < C$ classes, *i.e.* each client has more than one class. Furthermore, we assume that the domain shifts across the clients are negligible. Here, we use $K = 4$ clients and $C = 6$ classes to simulate the situation. We use the four batches (batch 2 - batch 5, with default batch splits) of ChestX-ray14 dataset to create four clients. Each client contains 10^3 partially labeled CXR images. The label statistics are presented in Table I. We also prepare 300 positive cases and 300 negative cases for each of six classes from other batches as an independent balanced test set.

2) Results: The numerical results are presented in Table II.

Impact of Partial Supervision Compared with the *Oracle*, the fully supervised baseline, all PSL methods (both centralized and decentralized) are negatively impacted, as PSL methods only receive partial labels. However, PSL methods can achieve competitive performance on certain diseases, even with partial labels, *e.g.* "Consolidation" and "Pneumothorax". This means that PSL methods do make a difference where there are limited training examples as they can extract meaningful information even without full supervision (with respect to certain classes).

Analysis of Centralized PSL Methods Recall that IML [2] is the centralized baseline that does not utilize any advanced techniques (*e.g.* pre-training, pseudo labeling, regularization). MixUp-PME [9] improves the overall performance of IML, while it slightly decreases the performance of several diseases compared to IML. We conjecture that the performance of MixUp-PME is class-dependent and can further be improved with sophisticated training tricks such as the ones described in [9]. FixMatch [42] performs even worse than IML as a state-of-the-art pseudo-label based *semi*-SL method. A similar phenomenon is also reported in [8], where pseudo labeling-based PSL methods are less robust with only limited labeled data, *i.e.* they fail to generate high-quality pseudo labels.

Impact of Decentralization Based on Table II, decentralization seems to have a positive impact on PSL: When comparing the centralized and decentralized methods, decentralized PSL methods outperform centralized PSL methods by a large margin. This is actually counter-intuitive, as under standard SL, decentralization usually slows the convergence rate and lowers the overall performance. As all methods are trained by SGD with random mini-batch sampling, we conjecture that centralization can amplify the negative impact of partial supervision. In a centralized environment, a random mini-batch could contain partially labeled examples from several partially labeled datasets. The partial supervision can cause inefficient training during back-propagation, especially when C is large. In contrast, decentralization slightly alleviates this issue. In each client, each mini-batch only contains relevant partial labels. This process can be more efficiently than centralized training for the model to extract transferable representations for only a few classes. Besides, in a centralized environment, when there are "dominating" classes (classes with more partial labels) or "easy" classes (classes with lower learning difficulty), the learning process will be inevitably influenced by such class imbalance. Meanwhile, decentralization ensures that local training is conducted independently and class imbalance is somewhat mitigated by model aggregation.

Comparison with Federated Baselines `FedAVG` and `FedMatch+` are the two best performing FPSL baselines considering average F1-score. While *self*-SL has played an important role in label-efficient learning, self-supervised pre-training shows different influences on `FedAVG` and `FedMatch`, a state-of-the-art federated *semi*-SL method. The federated pre-training slightly improves the performance of `FedAVG` on a few classes at the cost of decreased performance on other diseases. On the contrary, `FedMatch` significantly benefits from federated pre-training on all classes. As mentioned before, *semi*-SL suffers from label scarcity to produce high-quality pseudo labels. The results show that `FedMatch+` can leverage pre-training to deal with label scarcity. It is worth mentioning that `FedMatch+` (*i.e. self*-SL and *semi*-SL) does significantly improve the performance on several classes over the `FedAVG` baseline. However, similar to `FedSSP`, the performance of a few classes are negatively influenced. In contrast to standard image recognition tasks, such as CIFAR-10 and CIFAR-100 [43], which are commonly used in FL studies and assumed to have similar learning difficulties, medical images require additional consideration. In addition to the number of partial labels for each class, we hypothesize that the performance of `FedMatch+` is also determined by the learning difficulty of each class, which is not reflected in Sec. V-C.

Analysis of `FedPSL` `FedPSL`, while being based on `FedAVG`, outperforms all FPSL baselines by a large margin. Meanwhile, we also notice that `FedPSL` can even outperform *Oracle* on "Consolidation". To have a better understanding on the credit assignment of the two proposed modules, we further include ablation models that either include only the TDMA (task-dependent model aggregation) module or only the TADL (task-agnostic decoupling learning) module. These ablation models are denoted as `FedPSL` *w/ TDMA* and `FedPSL` *w/ TADL*, respectively. The two modules both improve the performance of `FedAVG`, with TADL having a larger impact. However, as shown in Algorithm 1, TDMA requires less computation to achieve better performance than TADL and is thus more computationally-efficient.

F. Federated System with One-Class Clients

1) Data Setup: In the second experiment, we consider a different scenario, where each client has $1 \leq C_k < C$ classes,

TABLE II: Quantitative evaluation of multi-label thoracic disease classification in a federated system **without** one-class clients. The reported number are the mean and standard deviation of average AUROCs over three random seeds. **Bold** denotes the highest number in each column (other than the *Oracle*). The column "**Average**" denotes the average AUROCs over classes, which is considered as the overall performance for a particular method.

	Method	Infiltration	Effusion	Atelectasis	Nodule	Consolidation	Pneumothorax	**Average**
Decentralized	FedAVG	0.648 ± 0.027	0.704 ± 0.015	0.690 ± 0.007	0.731 ± 0.021	0.814 ± 0.012	0.800 ± 0.015	0.731 ± 0.008
	FedProx	0.604 ± 0.049	0.719 ± 0.003	0.673 ± 0.010	**0.762** ± 0.037	0.819 ± 0.011	0.793 ± 0.014	0.728 ± 0.009
	FedSSP	0.618 ± 0.065	0.734 ± 0.015	0.697 ± 0.005	0.632 ± 0.008	0.799 ± 0.005	0.702 ± 0.014	0.697 ± 0.011
	FedMatch	0.676 ± 0.048	0.710 ± 0.044	0.717 ± 0.042	0.562 ± 0.016	0.761 ± 0.041	0.656 ± 0.030	0.680 ± 0.032
	FedMatch+	0.693 ± 0.008	**0.778** ± 0.008	**0.748** ± 0.009	0.610 ± 0.012	0.824 ± 0.007	0.761 ± 0.002	0.736 ± 0.003
Centralized	IML [2]	0.599 ± 0.064	0.711 ± 0.009	0.662 ± 0.026	0.570 ± 0.022	0.717 ± 0.015	0.614 ± 0.046	0.646 ± 0.010
	FixMatch [42]	0.600 ± 0.017	0.700 ± 0.016	0.637 ± 0.014	0.515 ± 0.013	0.756 ± 0.004	0.559 ± 0.025	0.628 ± 0.005
	MixUp-PME [9]	0.561 ± 0.029	0.778 ± 0.016	0.721 ± 0.020	0.586 ± 0.045	0.713 ± 0.040	0.596 ± 0.040	0.659 ± 0.013
Ours	FedPSL *w/ TDMA*	0.662 ± 0.017	0.757 ± 0.007	0.662 ± 0.064	0.719 ± 0.093	0.845 ± 0.009	0.799 ± 0.030	0.741 ± 0.009
	FedPSL *w/ TADL*	0.692 ± 0.018	0.765 ± 0.007	0.720 ± 0.067	0.731 ± 0.091	0.848 ± 0.012	**0.807** ± 0.015	0.757 ± 0.009
	FedPSL	**0.696** ± 0.017	0.771 ± 0.007	0.720 ± 0.059	0.738 ± 0.081	**0.848** ± 0.011	0.803 ± 0.015	**0.763** ± 0.011
Oracle		0.868 ± 0.010	0.907 ± 0.025	0.931 ± 0.015	0.792 ± 0.002	0.823 ± 0.009	0.809 ± 0.004	0.855 ± 0.010

TABLE III: Label statistics of positive cases for the experimental setup **with** one-class clients in the federated system. "0" denotes that the client does not have any partial labels for the disease, *i.e.* this disease is unlabeled.

Disease	Clients					
	1	2	3	4	5	6
Infiltration	141	0	0	0	0	0
Effusion	131	98	0	0	0	0
Atelectasis	78	91	121	0	0	0
Nodule	0	45	67	55	0	0
Consolidation	0	0	40	49	0	0
Pneumothorax	0	0	0	45	0	0
Tuberculosis	0	0	0	0	167	0
COVID-19	0	0	0	0	0	262

i.e. there can be one-class clients. In contrast to multi-class clients, the one-class client commonly collects data for a specific purpose, *e.g.* a less common disease which is unseen in other clients. For these less common diseases, the imaging protocols of these CXRs could be different from the common diseases, which leads to domain shifts (see Fig. 2). Thus, we further assume that domain shifts exist across the clients. Here, we use $K = 6$ clients and $C = 8$ classes to simulate the situation. We use the four batches (batch 2 - batch 5, with default batch splits) of the ChestX-ray14 dataset to create the first four clients, and use the Tuberculosis and COVID-19 datasets to create two one-class clients. Each client contains 10^3 partially labeled CXR images. The label statistics are presented in Table III. We also prepare 300 positive cases and 300 negative cases for each of eight classes from other batches as an independent balanced test set.

2) Results: Based on the empirical results in Table II, we choose the two best-performing federated PSL methods FedAVG and FedMatch+ as the baselines. The numerical results are presented in Table IV.

Impact of One-Class Clients When comparing the results for this experiment (Table IV) with Table II, we observe an interesting phenomenon: with new one-class clients added into the federated system, the *Oracle* tends to get lower performance on the six diseases in Table II. At first glance, this appears counter-intuitive, as usually, more labeled data should lead to better performance. Meanwhile, federated PSL methods all achieve robust overall performance ("**Average**") and can even outperform the *Oracle* on several classes (*e.g.* "Nodule" and "Tuberculosis"). A good example is "Tuberculosis". The CXRs of the Tuberculosis dataset are visually different from the CXRs of the Chest-Xray14 dataset, as shown in Fig 2. As the classes of the one-class clients are unseen in the other clients, the *Oracle* and PSL methods in fact have the same training data for 'Tuberculosis". However, the *Oracle* performs worse. This suggests that the feature extractor of the *Oracle* is negatively influenced by this newly added disease with obvious domain shift. This indicates that federated PSL methods are robust against the one-class clients and can even benefit from certain level of decentralization.

Analysis of FedPSL As in the previous experiment (Table II), FedPSL achieves the best overall performance and competitive performance on every single class. Notably, FedPSL outperforms all other methods, including *Oracle*, on "Tuberculosis" and "COVID-19", by a large margin. As these two classes are only present in two different clients, this further validates our discussion in Sec. V-C regarding the necessity of TDMA. Compared with the *Oracle*, the results further suggest that severe overfitting is a main challenge under data scarcity, where TADL is a more efficient solution than TDMA.

G. Ablation Studies

In addition to Sec. VII-E and Sec. VII-F, we further design a few ablation experiments to evaluate FedPSL under various situations, namely under different levels of data scarcity, with extreme clients, and under different levels of synchronization frequency. For simplicity, we use the data setup in Sec. VII-E.1, where the federated system has $K = 4$ clients and $C = 6$, *i.e.* without one-class clients. We choose FedAVG as the baseline in this section for its robust performance in previous experiments.

1) Sensitivity to Data Scarcity: In the first ablation experiment, we study the impact of data scarcity to FedPSL. Here, data scarcity describes the situation that each client only has access to small amounts of data. We denote that each client has n partially labeled examples (we sample the first n examples in each batch). The relation between the overall performance and n is depicted in Fig. 3. FedPSL shows obvious advantages over FedAVG when n is small, while the performance gain diminishes as n increases. This matches our expectations given the discussion in Sec. VI: TDMA and TADL are designed

TABLE IV: Quantitative evaluation of multi-label thoracic disease classification a federated system **with** one-class clients. The reported number are the mean and standard deviation of average AUROCs over three random seeds. **Bold** denotes the highest number in each column (other than the *Oracle*). The column "**Average**" denotes the average AUROCs over classes, which is considered as the overall performance for a particular method.

Method	Infiltration	Effusion	Atelectasis	Nodule	Consolidation	Pneumothorax	Tuberculosis	COVID-19	**Average**
FedAVG	0.624 ± 0.029	0.721 ± 0.025	0.701 ± 0.013	**0.791** ± 0.012	0.819 ± 0.013	0.785 ± 0.013	0.914 ± 0.010	0.837 ± 0.012	0.774 ± 0.011
FedMatch+	**0.693** ± 0.004	**0.773** ± 0.009	0.739 ± 0.009	0.664 ± 0.020	0.810 ± 0.004	0.781 ± 0.004	0.905 ± 0.003	0.780 ± 0.002	0.768 ± 0.005
FedPSL (*w/ TDMA*)	0.615 ± 0.010	0.752 ± 0.001	0.692 ± 0.017	0.736 ± 0.031	**0.836** ± 0.055	0.784 ± 0.032	**0.971** ± 0.008	0.832 ± 0.026	0.773 ± 0.010
FedPSL (*w/ TADL*)	0.653 ± 0.027	0.757 ± 0.027	0.745 ± 0.015	0.779 ± 0.015	0.821 ± 0.016	0.809 ± 0.021	0.941 ± 0.008	0.862 ± 0.028	0.799 ± 0.013
FedPSL	0.679 ± 0.027	0.765 ± 0.027	**0.756** ± 0.015	0.784 ± 0.014	0.834 ± 0.015	**0.819** ± 0.019	0.967 ± 0.008	**0.862** ± 0.028	**0.809** ± 0.013
Oracle	0.810 ± 0.025	0.873 ± 0.011	0.865 ± 0.017	0.733 ± 0.049	0.818 ± 0.013	0.788 ± 0.038	0.884 ± 0.053	0.833 ± 0.013	0.825 ± 0.020

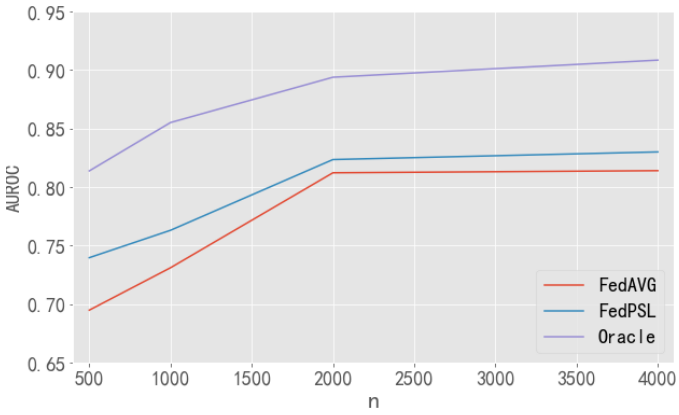

Fig. 3: Sensitivity of the overall performance (AUROC) to the size of the local dataset (n). The overall performance is measured by the mean AUROC over $C = 6$ classes in $K = 4$ clients.

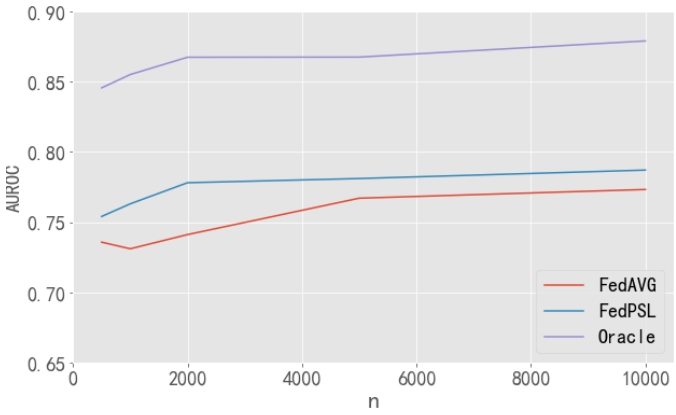

Fig. 4: Impact of extreme clients on the overall performance (AUROC). n denotes the size of the local dataset in the extreme clients, while each common client has 1000 partially labeled examples. Two out of four clients are the extreme ones. The overall performance is measured by the mean AUROC over $C = 6$ classes in $K = 4$ clients.

for data scarcity situations. As discussed in Sec. VI-B, with large-scale data, using TADL could lead to inefficient training. A potential solution is hybrid training (standard training for clients with large local datasets and TADL for clients with small local datasets), an exploration of which is considered beyond the scope of this work.

2) Sensitivity to Extreme Clients: In the second ablation experiment, we study the impact of extreme clients to FedPSL. In the previous experimental setups, we tend to assume that each client has access to the same number of samples. Here, instead, the *extreme* clients denote the ones that have significantly smaller or larger local datasets. While we follow the previous setup, we choose the second and the third clients to be our extreme clients. As each batch of the Chest-Xray14 dataset has 10^4 CXRs, each client can have up to 10^4 CXRs. Again, we use n to denote the size of the local datasets in the extreme clients. The relation between the overall performance and n is depicted in Fig. 4. FedPSL can achieve more robust performance that FedAVG. By comparing Fig. 3 and Fig. 4, we can also infer that, when the federated systems have roughly the same sizes of total data, the system with extreme clients has lower performance than the system without extreme clients.

3) Impact of Synchronization Frequency: An important hyperparameter in FL, which is often ignored, is the synchronization frequency. In practice, it is unlikely to exchange model parameters instantly (*e.g.* directly exchanging gradients in distributed machine learning [30]). We perform an ablation study on the synchronization frequency. We use E to denote the number of local training epochs before synchronization.

TABLE V: Impact of synchronization frequency on the overall performance (AUROC). The reported numbers are the mean and standard deviation of average AUROCs over three random seeds. The overall performance is measured by the mean AUROC over $C = 6$ classes. E denotes the number of local training epochs. **Bold** denotes the highest number in each column (other than the *Oracle*).

Method	E			
	10	20	50	100
FedAVG	0.731 ± 0.008	0.693 ± 0.010	0.645 ± 0.015	0.628 ± 0.011
FedPSL	**0.763** ± 0.011	**0.713** ± 0.024	**0.658** ± 0.015	**0.639** ± 0.013
Oracle	0.855 ± 0.010	0.855 ± 0.010	0.855 ± 0.010	0.855 ± 0.010

The results are summarized in Table V. In comparison with FedAVG, FedPSL is more robust against low synchronization frequency, while the performance gap diminishes as E grows up. However, FedPSL exhibits a higher standard deviation than FedAVG.

VIII. Conclusion

In this paper, we formulate and discuss the new problem of federated partially supervised learning (FPSL) for limited decentralized partially labeled medical images. We theoretically discuss the challenges of FPSL and present FedPSL, a simple yet robust solution to FPSL. We propose a task-dependent model aggregation module to address the aggregation weight assignment issue and a task-agnostic decoupling learning module based on meta-learning to address the local overfitting issue. Finally, we provide an empirical understanding of FPSL and our results indicate a new research direction in label-efficient learning with partial supervision.

REFERENCES

[1] O. Petit, N. Thome, A. Charnoz, A. Hostettler, and L. Soler, "Handling missing annotations for semantic segmentation with deep convnets," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 20–28.

[2] G. González, G. R. Washko, and R. S. J. Estépar, "Multi-structure segmentation from partially labeled datasets. application to body composition measurements on ct scans," in *Image Analysis for Moving Organ, Breast, and Thoracic Images*. Springer, 2018, pp. 215–224.

[3] Y. Zhou, Z. Li, S. Bai, C. Wang, X. Chen, M. Han, E. Fishman, and A. L. Yuille, "Prior-aware neural network for partially-supervised multi-organ segmentation," in *ICCV*, 2019, pp. 10 672–10 681.

[4] X. Fang and P. Yan, "Multi-organ segmentation over partially labeled datasets with multi-scale feature abstraction," *IEEE TMI*, 2020.

[5] G. Shi, L. Xiao, Y. Chen, and S. K. Zhou, "Marginal loss and exclusion loss for partially supervised multi-organ segmentation," *Medical Image Analysis*, p. 101979, 2021.

[6] Y. Xu, X. Xu, L. Jin, S. Gao, R. S. M. Goh, D. S. Ting, and Y. Liu, "Partially-supervised learning for vessel segmentation in ocular images," in *MICCAI*. Springer, 2021, pp. 271–281.

[7] J. Zhang, Y. Xie, Y. Xia, and C. Shen, "Dodnet: Learning to segment multi-organ and tumors from multiple partially labeled datasets," in *CVPR*, 2021, pp. 1195–1204.

[8] N. Dong, M. Kampffmeyer, X. Liang, M. Xu, I. Voiculescu, and E. Xing, "Towards robust partially supervised multi-structure medical image segmentation on small-scale data," *Applied Soft Computing*, p. 108074, 2022.

[9] N. Dong, J. Wang, and I. Voiculescu, "Revisiting vicinal risk minimization for partially supervised multi-label classification under data scarcity," in *CVPR*, 2022, pp. 4212–4220.

[10] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[11] European Commission, "General data protection regulation," 2016. [Online]. Available: https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en

[12] US Department of Health and Human Services, "Health insurance portability and accountability act," 2017. [Online]. Available: https://www.cdc.gov/phlp/publications/topic/hipaa.html

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

[14] S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi, "Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data," in *ISBI*. IEEE, 2019, pp. 270–274.

[15] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein *et al.*, "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[16] 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*, vol. 2, no. 6, pp. 305–311, 2020.

[17] P. Guo, P. Wang, J. Zhou, S. Jiang, and V. M. Patel, "Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning," in *CVPR*, 2021, pp. 2423–2432.

[18] K. V. Sarma, S. Harmon, T. Sanford, H. R. Roth, Z. Xu, J. Tetreault, D. Xu, M. G. Flores, A. G. Raman, R. Kulkarni *et al.*, "Federated learning improves site performance in multicenter deep learning without data sharing," *JAMIA*, vol. 28, no. 6, pp. 1259–1264, 2021.

[19] I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai *et al.*, "Federated learning for predicting clinical outcomes in patients with covid-19," *Nature Medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.

[20] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *NIPS*, 2014, pp. 19–27.

[21] G. Anandalingam and T. L. Friesz, "Hierarchical optimization: An introduction," *Annals of Operations Research*, vol. 34, no. 1, pp. 1–11, 1992.

[22] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*. PMLR, 2017, pp. 1126–1135.

[23] W. Zhuang, X. Gan, Y. Wen, S. Zhang, and S. Yi, "Collaborative unsupervised visual representation learning from decentralized data," in *ICCV*, 2021, pp. 4912–4921.

[24] F. Yu, A. S. Rawat, A. Menon, and S. Kumar, "Federated learning with only positive labels," in *ICML*. PMLR, 2020, pp. 10 946–10 956.

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

[26] N. Dong and I. Voiculescu, "Federated contrastive learning for decentralized unlabeled medical images," in *MICCAI*. Springer, 2021, pp. 378–387.

[27] C. Dwork, "Differential privacy," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2006, pp. 1–12.

[28] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[29] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[30] W. Dai, Y. Zhou, N. Dong, H. Zhang, and E. Xing, "Toward understanding the impact of staleness in distributed machine learning," in *ICLR*, 2019.

[31] 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.

[32] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[33] T. G. Dietterich, "Ensemble methods in machine learning," in *International Workshop on Multiple Classifier Systems*. Springer, 2000, pp. 1–15.

[34] Z.-H. Zhou, "Ensemble learning," in *Machine learning*. Springer, 2021, pp. 181–210.

[35] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of operations research*, vol. 153, no. 1, pp. 235–256, 2007.

[36] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *CVPR*, 2017, pp. 2097–2106.

[37] T. Rahman, A. Khandakar, M. A. Kadir, K. R. Islam, K. F. Islam, R. Mazhar, T. Hamid, M. T. Islam, S. Kashem, Z. B. Mahbub *et al.*, "Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization," *IEEE Access*, vol. 8, pp. 191 586–191 601, 2020.

[38] M. E. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. Al Emadi *et al.*, "Can ai help in screening viral and covid-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132 665–132 676, 2020.

[39] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021, pp. 15 750–15 758.

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017, pp. 4700–4708.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[42] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *NIPS*, vol. 33, 2020, pp. 596–608.

[43] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.