Federated Partially Supervised Learning for Decentralized Medical Images

***Abstract*— Medical data regulations tend to To enhance the data government situation in privacy-critical infrastructure, decentralized learning has played an important role in the medical domain. As a trade-off, decentralization can also limit the effectiveness of partially supervised learningdue to the lack of access to decentralized data, leading to a bottleneck for label-efficient learning. As a remedy, this work formulates and discusses a new learning problem *federated partially supervised learning* (FPSL) for decentralized medical images with partial labels. We illustrate the negative impact of decentralized partially labeled data on neural networks and the necessity of studying FPSL via a set of multi-label thoracic disease classification tasks. Motivated by the empirical observations in this paper, an examplar task of FPSL, namely, *federated partially supervised learning multi-label classification*. To tackle label scarcity and class imbalance, two major challenges of FPSL, we propose a simple yet robust FPSL framework, `FedPSL`. `FedPSL` is a two-stage training framework, where the first stage is a federated unsupervised pre-training stage to learn class-agnostic representations. The second stage is a novel federated consistency regularization training stagewhich utilizes partial labels to learn task-dependent representations. To mitigate overfitting caused by both non-independent and identically distributed (non-IID) data and partial supervision in FPSL, we present two modules, a consistency regularization local training module and a divergence-aware global aggregation module. We use the consistency regularization module to alleviate local label scarcity and use the divergence-aware partially supervised training stage, where each client consists of a feature extractor and a prototype-based multi-label classifier. We further propose a local consistency regularization training module to generate high-confidence pseudo labels and a global aggregation module to compensate for the cross-site mitigate class imbalance. The empirical results not only indicate that FPSL is an under-explored problem with practical values but also show that the proposed `FedPSL` can achieve robust performance against baseline methods on non-IID data challenges data challenges such as label scarcity and class imbalance. 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) [?], [?], [?], [?], [?], [?], [?], [?] [?], [?], [?], [?], [?], [?], [?], [?], [?] has become an emerging 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* [?] or *partial labels problem* [?] 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) [?]: 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.

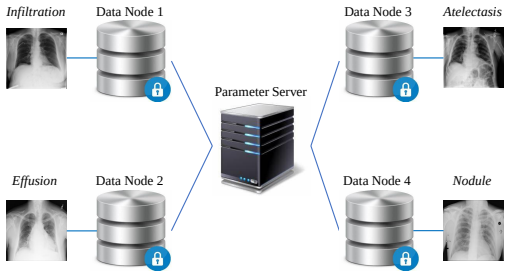

Fig. 1: Illustration of FPSL for a multi-label classification task on chest X-ray images. In each client (data node), each image is partially labeled for only one thoracic disease, *e.g.* we only know whether each image in the first data node has *infiltration* but have no knowledge on the other three diseases. Due to data regulations, 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 four partially labeled datasets stored in different data nodes to train the model of interest in the parameter server.

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. Thus, it is natural to think about the connection between PSL and *federated learning* (FL) [?]. 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 [?], [?], [?], [?], [?], [?]. 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. Moreover, in Sec. VI, we experimentally show that FPSL is different from existing learning paradigms, such as supervised learning, semi-supervised learning, and self-supervised learning[1], and is an under-explored problem of interest.

It is worth noting that ~~FPSL for decentralized medical images can not be addressed by a simple~~ a direct combination of FL and PSL ~~, as it differs from these two problems in two aspects~~does not provide a robust solution to the problem of interest. Firstly, ~~in contrast to the standard fully-supervised setting in FL[2], FPSL suffers from label scarcity. That is to say, in addition to the common challenge of~~ the federated setup poses a non-trivial barrier for the implementation of some PSL methods. For example, VRM-based PSL methods [?], [?] require access to the training data in a centralized fashion and label propagation-based PSL methods [?] involve iterated training over each partially labeled dataset to generate pseudo labels, 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 [?], [?]. Thirdly, in FL, the data are assumed to be non-independent and identically distributed (non-IID)~~data, FPSL has to consider the effect of class imbalance caused by the partial labels in each client when conducting global aggregation. This raises questions regarding the robustness of centralized PSL methods that use complex training procedures [?] or network architectures [?]. Secondly, data regulations in the medical domain pose a new challenge to both FL and PSL~~. 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 existing PSL methods have tried to tackle these challenges. Fourthly, an additional consideration to take into account is the data regulations. Here, data regulations refer to situations where the data stored in the client are not allowed to be transferred to the server or other clients. A detailed explanation 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 [?] and US Health Insurance Portability and Accountability Act [?]). ~~With this new constraint, some of the previous PSL methods will be infeasible to implement. For example, [?] utilizes partial labels to generate vicinal labels for supervised learning (SL), thus can only be implemented as centralized training.~~

~~A key problem that FPSL~~aims to address is the ~~overfitting caused by~~ As the first study in FPSL, a primary goal in this work is to demonstrate the negative impact caused by decentralized partially labeled data. Specifically, we aim to provide an empirical understanding on the effects of label scarcity and ~~non-IID data, which has not been addressed in previous PSL studies [?], [?], [?], [?]. In particular, the learning process can be easily dominated by a few major classes when class imbalance~~exists ~~[?],~~ class imbalance under the federated setup, which both can lead to overfitting. Here, the term "label scarcity" refers to the situation that only limited partial labels are available in the clients. The term "class imbalance" has two meanings: the classes with more partial labels ~~will contribute more to the learned representations, which will lead to overfitting. One such~~ can dominate the learning process and 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~~is multi-label classification (MLC), which we use to illustrate the challenge of FPSL with affordable computational resources~~. 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 ~~[?]~~ [?], [?] or data augmentation [?]~~) and normally suffers from the class imbalance caused by long-tailed distributions~~.

To fill the aforementioned methodological gaps, we present a simple yet robust FPSL framework `FedPSL`. A direct consequence of label scarcity and class imbalance is overfitting. With limited labels, it is difficult to learn a robust feature extractor or learn to generate reliable pseudo labels directly. From the perspective of MTL, the learning process can also be dominated by a few sub-task with relatively more labels. Moreover, the design of multi-label classifier should also comply to the FL setup. To this end, `FedPSL` ~~consists of two training stages, which are designed to mitigate label scarcity efficiently~~utilizes a prototype-based multi-label classifier, where a mathematical link can be built between the prototypes and the weights of a standard linear classifier. In contrast to a standard classification layer [?], a prototype-based classification layer [?] is more robust against label scarcity and class imbalance. There are two training stages when optimizing the feature extractor and the prototypes. The first training stage is a federated unsupervised pre-training stage, where we leverage *self-supervised learning* (*self*-SL) to learn robust and transferable *class-agnostic* representations from unlabeled data (*i.e.* in the first training stage, all partially labeled data are deemed as unlabeled data). ~~As unsupervised pre-training has been demonstrated to achieve competitive performance compared to supervised pre-training in representation learning on benchmark tasks [?], we use *self*-SL to alleviate overfitting caused by class imbalance~~Specifically, we pre-train the feature extractor of the classifier of interest. The second training stage is a novel federated ~~*consistency regularization* (CR)~~ partially supervised training stage, which consists of two modules to facilitate the federated training with partial labels. The first module is a local training module that generates pseudo labels based on ~~CR. CR [?]~~consistency regularization [?], which was originally proposed to address the semi-supervised multi-class classification, which aims to enforce an agreement between the distributions of the pseudo labels generated for two augmented

[1]We use the terms "self-supervised learning" and "self-supervised representation learning" interchangeably in this work.

[2]~~For simplicity, we use FL to denote the federated learning task with standard supervised loss in this paper, unless stated otherwise.~~

views of the same unlabeled image. In this work, we adapt ~~CR for the partially supervised MLC~~ consistency regularization for the *partially supervised multi-label classification* [?] task for the first time. Without any fully labeled data, the proposed local module learns to produce pseudo partial labels with high confidence based on the knowledge learned from local ground truth partial labels and knowledge transferred from the other ~~data nodes. It is worth mentioning that the CR training here is easy to implement as it shares a similar workflow with the unsupervised pre-training in the first stage, which is illustrated in Fig. ??.~~ clients. We also leverage an alternative training strategy between the feature extractor and prototypes in each client to ensure smooth updates. The second module is a ~~divergence-aware aggregation moduledesigned to attenuate cross-site class imbalance in FL. Unlike the standard setup of FL~~prototype-based aggregation module, where the ~~weight for each local model is decided by the relative size of data in each client, both the amount of *effective* partial labels and the~~ aggregation mechanism is based on the distance metric between the local prototypes and the global prototypes to avoid large *weight divergence* [?] ~~between the local model and global model in each client are considered in the aggregation step. We theoretically and empirically analyze the advantages of this module over FedAVG [?], a seminal robust baseline for non-IID data~~and thus improve the FL performance.

As the first study of FPSL on decentralized medical images, we evidence the contributions of this work by assessing `FedPSL` in a simulated exemplar ~~PSL task,~~ task, federated partially supervised multi-label classification. We evaluate `FedPSL` against strong baselines in terms of both performance and robustness under various data challenges such as ~~class imbalance and data scarcity~~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 ~~FPSL framework for~~ framework for federated partially supervised multi-label classification.
2) We empirically demonstrate that federated unsupervised pre-training is a promising direction for FPSL.
3) We propose a novel federated ~~CR training pipeline for FPSL including a divergence-aware~~ partially supervised training pipeline including a prototype-based aggregation module, which ~~can compensate for the cross-node class imbalance~~ aims to smooth the FL process and improve the overall performance.
4) We show initial evidence that FPSL is an under-explored problem compared with existing learning paradigms and offer the community the first benchmark ~~MLC task under FPSL~~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 describes the proposed solution in detail. Sec. VI describes the proposed benchmark tasks and provides experimental results and analysis. Section VII summarizes this work.

II. Related Work

A. Federated Learning

As an emerging research area, there are limited FL studies related to this work. Three related areas are federated unsupervised representation learning (FURL) [?], federated *positive-unlabeld* (PU) learning [?], and federated semi-supervised learning (FSSL) [?]. FedU [?] discuss a divergence-aware update mechanism for FURL. Different from `FedPSL`, the divergence-aware module of FedU only considers the local updates instead of global aggregation. As a federated extension of PU learning, FedAwS [?] 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, FSSL is another related domain to FPSL. FSSL method FedMatch [?], for instance, also adopts a CR-based pseudo-labeling training strategy. However, compared to `FedPSL`, FedMatch has a completely different problem formulation, which enables the usage of fully labeled data, and does not utilize the link between instance discrimination-based *self*-SL and CR.

B. Partially Supervised Learning

Recently, there have been efforts made to utilize multiple partially labeled datasets in the medical domain. ~~[?] requires a fully-labeled dataset to learn the prior to perform expectation-maximization iterations. [?], [?] both rely on complex neural networks for centralized training . [?] proposes a marginal loss and an exclusion loss for multi-class segmentation while a fully-labeled dataset is required in the training. These methods do not directly solve the data scarcity problem and still rely on large amounts of data (even the fully labeled ones). As a comparison, VLUU [?] addresses the label scarcity issue by generating vicinal labels based on human structure similarity. However, VLUU can only work in a centralized training environment. As all these~~ However, none of these methods are designed for the situation that the partially labeled datasets are decentralized. [?], [?] address the partial labels issue by generating vicinal labels based on human structure similarity, which can only be implemented in a centralized training environment. [?], [?] 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 [?] have iterated training procedure, 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*. With small-scale local training data in each client, PSL methods with complex network architectures or training procedures [?], [?] normally perform much worse in a federated environment than the counterparts that have access to the large-scale training data in a centralized environment [?]. As existing PSL methods struggle in the problem formulation defined in Sec. ??IV), it is meaningful to develop a FPSL frameworkstudy the problem of FPSL and develop a robust solution.

III. Problem SetupPreliminaries

A. 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.[2] 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}})| \subseteq |\mathbb{S}(y_{\text{full}})|$$|\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 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 C K sub-datasets where each sub-dataset contains label information of only one classa few classes, *i.e.* $\mathcal{S} = \bigcup_k^C \mathcal{S}_k$, where $\mathcal{S} = \bigcup_k^K \mathcal{S}_k$. Here, $\mathcal{S}_k = \{(x_i^k, y_i^k)\}_{i=1}^{|\mathcal{S}_k|}$ denotes the partially labeled dataset for class $k \in \mathcal{C}$. Note, the assumption here is the most representative case as all other cases are trivial extensions [?]. For example , each partially labeled image could have more than one class annotated. 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* [?], each sub-dataset is only annotated for a true subset of $\mathcal{C}$.

A. Federated Partially Supervised Learning

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

IV. Problem Setup

Now, we formulate the problem of *federated partially supervised learningmulti-label classification*(FPSL). Without loss of generality, we consider the most representative case where $K = C$, each data node (client) only has partial labels for a unique class. Let , an exemplar task of FPSL. Analogous to Sec. III-.1, we have K clients (data nodes) in a federated system and $\mathcal{S}_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$ denote the denotes the partially labeled dataset stored in node $k \leq K$, where $n_k = |\mathcal{S}_k|$ and y_i^k is the partial label of x_i^k with respect to class $k \in \mathcal{C}$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*:

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

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) [?]. In addition to the standard setup of FL, the data regulations in the medical domain pose a new 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) [?] 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 due to regulations. 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. Thus, the discussion on federated communication and privacy-preserving techniques is beyond the scope of this work.

V. Method

In this section, we present the proposed `FedPSL` framework, which consists of two stages. The first stage is a federated unsupervised pre-training stage (Sec. V-A). We use *self*-SL to learn meaningful and transferable representations across clients without any labels, which can be efficiently used to mitigate the label scarcity and class imbalance and also provide a strong initialization for the second stage. Given the pre-trained feature extractor and prototypes extracted by the feature extractor, an initial prototype-based multi-label classifier can be built. The second stage is a federated consistency regularization partially supervised training stage, which consists of a local consistency regularization training module (Sec. ??V-B.1) and a divergence-aware novel global aggregation module (Sec. ??).

Unsupervised Pre-Training. The first stage is based on self-supervised learning via an instance discrimination pretext task. While both the encoder (green) and predictor MLP (light green) are trained in the first stage, only f_θ will be leveraged in the next training stage. Consistency Regularization Training. The second stage is based on semi-supervised learning. The figure only illustrates the *self-training* procedure with pseudo labels. Illustration of local training modules for (a) the federated unsupervised pre-training stage (Sec. V-A) and (b) the federated consistency regularization training stage (Sec. V-B). Both training stages leverage two-view *self-training* to maximize mutual agreement between two views. V-B.2). The local training module can utilize ground truth partial labels and further generate high-confidence pseudo partial labels to fine-tune the feature extractor and improve the quality of prototypes. The global aggregation module is designed to avoid large *weight divergence* [?] to stabilize the training process. An end-to-end training pipeline is presented in Sec. V-B.3.

A. Stage 1: Federated Unsupervised Pre-Training

In contrast to standard SL, PSL has sparser annotations given the same amount of training data , for a multi-label classification task formulated under Sec. ??, there are only $\frac{1}{C}$ of annotations compared to the fully labeled counterpart. Under the federated setup, the decentralization will exacerbate the class imbalance caused by the different number of partial labels in each client. Thus, a straight-forward local partially supervised training might fail to grasp the features in regard to unlabeled classes in that client. As we assume that each client only has a few classes labeled, this can lead to severe overfitting for other clients.

So far, unsupervised pre-training has been demonstrated to achieve competitive performance compared to supervised pre-training in representation learning on benchmark tasks [?], [?]. To maximally utilize the available training dataand mitigate the overfitting caused by limited partial labels, either partially labeled or unlabeled, we leverage *self*-SL to learn transferable class-agnostic representations across data nodes during the first stage.

1) Local Training: We use SimSiam [?], a state-of-the-art In the first stage, all available data will be considered as unlabeled as no labels will be involved in the training. As the focus of this work is FPSL, we simply make the first stage *framework-agnostic*. In other words, any *self*-SL framework, as the backbone framework in each data node. We choose SimSiam for the following two advantages: (i) SimSiam does not require a large batch size [?] or a dynamic dictionary [?] for unbiased estimation [?]; and (ii) SimSiam does not require an additional copy of model weights ([?]). Thus, SimSiam allows for a lightweight implementation.

SimSiam formulates the pretext task based on instance discrimination, by maximizing the similarity between two different views of the same instance. Given an input image x and two augmentations $\tau \sim \chi$, $\tau' \sim \chi$ that are sampled from an image augmentation distribution χ. $\tau(x)$ and $\tau'(x)$ can be generated as two different views of x and used as the inputs for the same network, separately. As the original data augmentation policy in [?] is not suitable for medical images, we follow the data augmentation policy proposed in [?].

As the data nodes share the same network architecture, we use θ to denote the parameters θ_k in node k. SimSiam has two networks: an encoder f_θ and a predictor h_θ[3]. Two augmented views $\tau(x)$ and $\tau'(x)$ are both processed by f_θ. However, the predictor h_θ is applied on one view and a *stop-gradient operation* [?], [?] is applied on the other view. Concretely, let $p_1 = h_\theta \circ f_\theta(\tau(x))$ and $z_2 = f_\theta(\tau'(x))$. The pretext task is to minimize the negative cosine similarity between the embeddings of the two views:

$$\mathcal{L}_{sim}(\tau(x), \tau'(x)) = -\frac{p_1}{\|p_1\|}\frac{z_2}{\|z_2\|},$$

where $\|\cdot\|$ is the l_2-norm. Following a similar definition, a symmetrized loss [?], [?] can be defined as

$$\mathcal{L}_\theta = \frac{1}{2}\mathcal{L}_{sim}(\tau(x), \tau'(x)) + \frac{1}{2}\mathcal{L}_{sim}(\tau'(x), \tau(x)).$$

The overall framework for unsupervised pre-training is illustrated in Fig. ??.

1) Global Aggregation: The model of interest is f_θ and its parameter set. At the beginning of the federated training , -SL framework should be feasible. In each federated training round, let $\{\theta_k\}_{k=1}^K$ be the model weights uploaded to the global model weights θ_0 are randomly initialized in the PS and PS by K copies of θ_0 are distributed to each data node as $\{\theta_k\}_{k=1}^K$, the data nodes are fully synchronized with the PS . Then, node k updates clients, where θ_k by training on $\mathcal{S}_k$ independently. After the same number of local epochs, is updated by performing locally *self*-SL (*e.g.* instance discrimination [?], [?], [?], [?]). $\{\theta_k\}_{k=1}^K$ are aggregated into θ_0 in the PS and the data nodes are fully synchronized with the PS. Given the metadata of $\{\mathcal{S}_k\}_{k=1}^K$, we simply aggregate $\{\theta_k\}_{k=1}^K$ by FedAvg [?], :

$$\theta_0 = \sum_k \frac{n_k}{n}\frac{n_k}{\sum_k n_k}\theta_k, \tag{2}$$

where $n = \sum_k n_k$.

The federated unsupervised pre-training stage is depicted in Algorithm 1. The process $n_k = |\mathcal{S}_k \cup \mathcal{U}_k|$ is the count of local training and global aggregation is repeated for a pre-defined number of rounds (T)examples. Note, only model parameters $\{\theta_k\}_{k=0}^K$ and metadata are exchanged between the PS and the data nodesthe metadata $\{n_k\}_{k=1}^K$ will be uploaded to the PS along with the model weights.

Input: θ_0^0, $\{\mathcal{S}_k\}_{k=1}^K$, T **Output:** θ_0^T Federated Unsupervised Pre-Training Stage. We use t to denote the t^{th} round of federated training. Each round consists of a few local epochs. T is the total number of rounds. We use θ_k^t to denote the model weights stored in node k at the t^{th} round.

[3]In fact, the encoder here consists of a feature extractor and a projector. The feature extractor is the model of interest (ResNet18 [?]). The projector is a *multi-layer perceptron* (MLP) for dimensionality reduction. The predictor is another MLP. After the unsupervised pre-training, only the feature extractor is kept, both the projector and the predictor are discarded. The interested reader is referred to [?] for further details.

B. Stage 2: Federated ~~Consistency Regularization~~ Partially Supervised Training

~~Similar to Sec. V-A, the federated consistency regularization (CR) training stage also has a local training module and a global aggregation module. We will present the second stage in four~~ The second stage is presented in three steps. Firstly, ~~in~~ we describe the local training module ~~, we adapt CR training to PSL in Sec. ??. Secondly, we discuss the necessity of two-stage training in Sec. ??, as a complement to Sec. ??. Thirdly, we introduce the~~ from the perspective of MTL. Secondly, we propose a novel global aggregation module ~~for the second stage in Sec. ??, including a novel divergence-aware method. Fourthly, we theoretically analyze the mechanism of the proposed divergence-aware aggregation modulein Sec. ??.~~ based on prototypes. Thirdly, we propose an alternative training strategy which can further improve the local training module.

1) ~~Consistency Regularization~~ Local Training with Partial Labels: ~~Given node k and S_k, only class k (or task k from a multi-task learning perspective [?]) is annotated.~~ Let f_{θ_k} be the feature extractor ~~inherited from~~ stored in the k^{th} client. f_{θ_k} is initialized by the federated unsupervised pre-training described in Sec. V-A. ~~In addition to~~ Let g_ϕ be the multi-label classifier on top of the feature extractor ~~, we introduce~~ f_{θ_k}. From a MTL perspective, g_ϕ can be decomposed into C binary ~~predictors $\{g_{\phi_{k,j}}\}_{j=1}^{C}$ for C,~~ classifier. Motivated by this, we construct a prototype-based multi-label classifier. Given image x, $f_{\theta_k}(x)$ is the extracted feature vector. Then, for class $c \in \mathcal{C}$, we define the *prototype* [?] of class c in client k as

$$\mu_k^c = \text{norm}(\frac{1}{|\mathcal{S}_k^c|} \sum_{x \in |\mathcal{S}^c|} f_\theta(x)), \tag{3}$$

where ~~$g_{\phi_{k,j}}$ denotes the binary prediction head for class $j \in \mathcal{C}$ in node~~ $\text{norm}(v) = \frac{v}{\|v\|}$ is a normalization operator and $\mathcal{S}_k^c$ denotes the set of images that are labeled as class c. If $c \in \mathcal{C}_k$, *i.e.* client k ~~. For example, for a multi-label classification problem, $g_{\phi_{k,j}} \circ f_{\theta_k}$ is a binary classifier for~~ has partial labels for class ~~j~~c, the initial $\mathcal{S}_k^c$ just contains the images with ground truth partial labels with respect to c. We use the unit vector following the *principle of maximum entropy* [?] because we have no prior knowledge on class c. $\mathcal{S}_k^c$ is updated dynamically as it can also contain images with high confidence pseudo labels with respect to c, which will be discussed below.

~~For class k, the supervised training~~ Let $\phi \in \mathbb{R}^{d \times C}$ denote the weight matrix of the linear classifier g_ϕ, where d is the dimension of the feature vector. The model prediction of image x can be formulated as $g_{\phi_k} \circ f_{\theta_k} = {\phi_k}^T \cdot f_{\theta_k}(x)$. The probability scores for C classes are

$$p(x) = \sigma({\phi_k}^T \cdot f_{\theta_k}(x)), \tag{4}$$

where σ is the sigmoid function which applies to each element of the prediction vector ${\phi_k}^T \cdot f_{\theta_k}(x)$. Mathematically, ${\phi_k}^T \cdot f_{\theta_k}(x)$ is to have a dot product between each column vector of ϕ_k and the feature vector $f_{\theta_k}(x)$. If we use C prototype vectors as C column vectors of ϕ_k, we are using cosine similarity to measure the similarity between $f_{\theta_k}(x)$ and each prototype vector.

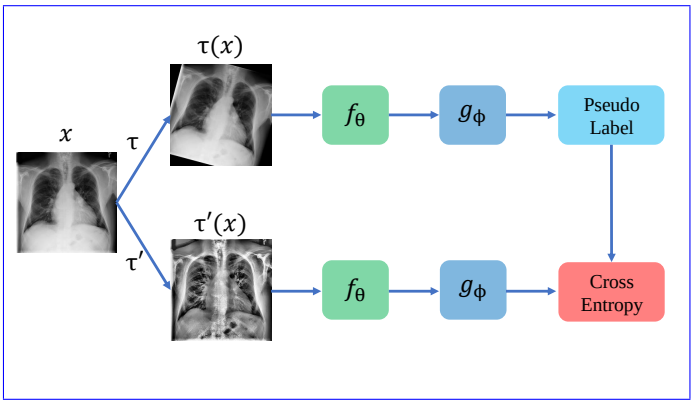

Fig. 2: Consistency Regularization Training.

With non-empty set $\mathcal{S}_k^c$ and the model $g_{\phi_k} \circ f_{\theta_k}$, the training for class c in client k is performed by minimizing the binary cross-entropy loss

$$\mathcal{L}_k^c = -\frac{1}{|\mathcal{S}_k^c|} \sum_{x \in \mathcal{S}_k^c} y^c p^c(x) + (1 - y^c) \log(1 - p^c(x)), \tag{5}$$

where ~~y_i^k denotes the label of x_i~~ y^c denotes the ground truth or pseudo partial label of x with respect to class ~~k and $p_k(\tau(x)) = g_{\phi_{k,k}} \circ f_{\theta_k}(\tau(x_i))$ is the predicted probability of $\tau(x)$ for~~ c.

As each client has only partially labeled data, $\mathcal{S}_k^c$ could be small for some $c \in \mathcal{C}_k$ or empty $\forall c \notin \mathcal{C}_k$. In addition to utilizing prototypes downloaded from the PS, it is also beneficial to generate pseudo labels to enlarge $\mathcal{S}_k^c$ to improve the quality of prototypes. After defining the prototypes and optimization object, we now get back to how to update $\mathcal{S}_k^c$. Theoretically, any *semi*-SL methods that can generate high confidence pseudo labels could work. In this work, we adopt *consistency regularization* training [?]. We adapt FixMatch [?], a state-of-the-art method, to the partially supervised multi-label classification problem for the first time. For class ~~k.~~

~~For class $j \neq k$, we~~ $c \notin \mathcal{C}_k$, the image x in client k does not have ground truth partial labels for class c. We first generate two augmented views $\tau(x)$ and $\tau'(x)$ for each image x. ~~Similar to Sec. ??, we aim to maximize the agreement between the predictions between two views~~, as shown in Fig. 2. The pseudo label for class ~~j~~ c is defined as

$$\hat{y}_{-}^{j\,c} = \begin{cases} 1, & \text{if } p^c(\tau(x)) \geq \alpha \\ 0, & \text{if } p^c(\tau(x)) \leq 1 - \alpha \end{cases}, \tag{6}$$

where ~~$p_j(\tau(x)) = g_{\phi_{k,j}} \circ f_{\theta_k}(\tau(x_i))$ and~~ $0 < \alpha < 1$ is a threshold value. The images with undefined pseudo labels (*i.e.* ~~$1 - \alpha < p_j(\tau(x)) < \alpha$~~ $1 - \alpha \leq p_c(\tau(x)) \leq \alpha$) will not be used in the training. As shown in [?], [?], α should be large enough to generate a pseudo label with high confidence or low uncertainty (*e.g.* $\alpha = 0.95$ in this work following [?]).

Analogous to Eq. , we minimize

$$\mathcal{L}_{pseudo} = -\frac{1}{n_k}\sum_j\sum_i y_i^j p_j(\tau(x)) + (1-y_i^j)\log(1-p_j(\tau'(x))).$$

The final optimization goal is to minimize the sum of Eq. and Eq.

$$\mathcal{L}_\phi = \mathcal{L}_{sup} + \lambda\mathcal{L}_{pseudo},$$

where λ is a hyperparameter which controls the weight of $\mathcal{L}_{pseudo}$ (we set $\lambda = 1$ following [?]). The overall framework for CR training in each data node is illustrated in Fig. **??**. We will describe the proposed global aggregation module in Sec. **??**Then, we aim to maximize the agreement between the prediction vector on $\tau'(x)$ and the pseudo label of $\tau(x)$ $\forall c \notin \mathcal{C}_k$.

2) Necessity of Two-Stage TrainingGlobal Aggregation: Again, assume only class k is annotated for node k. Without the federated unsupervised pre-training stage, the learning process could be easily dominated by the term $\mathcal{L}_{sup}$ when jointly optimizing θ and ϕ via minimizing Eq. , especially in the early stage of the training . With a large value of α, there will be much fewer pseudo labels than ground truth labels. This situation will inevitably lead to f_{θ_k} being overfitted to class k and further influence the training for $\{g_{\phi_{k,j}}\}_{j\neq k}$ in node k. Given the limited partial labels, where the majority of labels will be negative, f_{θ_k} will learn less meaningful representations (f_θ is overfitted to the negative cases) for the downstream tasks. Moreover, for non-IID datasets stored in different nodes, there could be domain shifts across data nodes (f_{θ_j} is overfitted to node j) At the end of each local training round t, $\{\theta_k\}_{k=1}^K$ and $\{\phi_k\}_{k=1}^K$ are uploaded to the PS. It is worth mentioning that $\{\phi_k\}_{k=1}^K$ are both model weights and metadata. Thus, we utilize the first federated unsupervised pre-training stage to mitigate overfitting.

One might argue that the federated mechanism (to be introduced exchanging $\{\phi_k\}_{k=0}^K$ between the PS and the clients does not violate the data regulations in Sec. **??**) could alleviate the overfitting by aggregating knowledge from other nodes. Note, however, when the node is trained with local data independently, the overfitting still exists. In addition, relying on federated aggregation alone is inefficient for two reasons. First, the updates from other nodes are delayed updates [?] (real-time updates). Second, the communication costs in FL are non-trivial.

3) Global Aggregation: The training in the second stage consists of a warm-up phase (Phase I) and a regular phase (Phase II), which have different global aggregation methods.

In **Phase I**, only the partial labels are utilized for supervised training in each data node, only $\phi_{k,k}$ is trained for node IV. We use two different aggregation mechanisms θ and ϕ. As θ_k represents the general knowledge learned from client $k$following , we simply aggregate $\{\theta_k\}_{k=1}^K$ with Eq. . This can be understood as warm-up training to learn reasonable priors for the prediction heads. This design is motivated by the fact that FPSL tasks commonly do not have fully labeled images (*semi*-SL [?]) and are more prone to overfitting caused by the class imbalance in each datanode2 where

$$n_k = \sum_{c\in\mathcal{C}} |\mathcal{S}_k^c|. \tag{7}$$

In contrast to θ_k, ϕ_k is prone to be influenced by local data. Note, because each node only has partially labeled data, minimizing Eq. at the beginning will not only learn unreliable prediction heads for classes with no labelsbut also introduce noise to f_θ. This design distinguishes this work from standard CR. In the warm-up training, the aggregation of the encoder f_θ simply follows Eq. . For the prediction heads, we set

$$\phi_{0,j} = \phi_{j,j},$$

for class $j \in \mathcal{C}$ ϕ can be decomposed into prototype vectors: $\phi_k = \{\mu_k^c\}_{c=1}^C$. The quality of ϕ is determined by the quality of prototype vectors, which is further influenced by local partial labels. It is highly possible that the prototypes of the same class exhibit large divergence across different clients. It is critical to ensure the stability when updating the prototypes in the PS.

During **Phase II**, both ground truth labels and pseudo labels are used in the training following Take class c as an example. Let $\mu_{0,t}^c$ be the global prototype of class c kept from last training round t and $\mu_{0,t+1}^c$ be the updated prototype of class c to be synchronized to K clients at the beginning of training round $t+1$. We aggregate $\{\mu_k^c\}_{k=1}^K$ with Eq. . We propose a *divergence-aware aggregation module* (DAAM) based on two motivations. First, given a class, the weight assignment should reflect the contributions of the number of effective labels used in the training. Here, the effective label denotes the ground truth or pseudo partial label for a specific class used in 2:

$$n_k = \frac{\exp^{\text{sim}(\mu_{0,t}^c, \mu_k^c)}}{\sum_j \exp^{\text{sim}(\mu_{0,t}^c, \mu_j^c)}}, \tag{8}$$

where $\text{sim}(\cdot,\cdot)$ is similarity measure (*e.g.* cosine similarity) between two vectors. $\text{sim}(\cdot,\cdot)$ can be interpreted as measuring the *weight divergence* [?], [?]. Eq. . Second, the weight assignment should also take the weight divergence caused by cross-site class imbalance into consideration. Intuitively, a class (or node) with a larger weight divergence implies the model learns more knowledge for this class (or node). Thus, we want to assign a small weight to the model with large weight divergence to balance the learning process. 8 implies that the local prototype with smaller weight divergence against the global prototype should be assigned larger weight in the global aggregation, which ensures a smooth update process.

We define the divergence-aware weight matrix A as:

$$A_{j,m} = \frac{\frac{n_{j,m}}{\|\phi_{j,m}^t - \phi_{0,m}^{t-1}\|_2^2}}{\sum_k \frac{n_{k,m}}{\|\phi_{k,m}^t - \phi_{0,m}^{t-1}\|_2^2}},$$

where $n_{j,m}$ denotes the number of ground truth or pseudo labels of class m in node j, $\|\cdot\|_2^2$ is the *Frobenius* norm that measures the weight divergence, $\phi_{j,m}^t$ denotes the weights

Algorithm 1 Federated Consistency Regularization Training Stage. We use t to denote the t^{th} round of federated Local training . Each round can consist of a few local epochs. T_w is the number of warm-up rounds. T is the total number of rounds. We use ϕ_k^t to denote the model weights stored in node k at the t^{th} round. The update and aggregation of θ are detailed in Sec. **??**, thus omitted here procedure for simplicityclient k.

Input: ϕ_0, $\{\mathcal{S}_k\}_{k=1}^K$, T_w, T $\mathcal{S}_k^c$: Labeled image set of class c
Output: ϕ_0^T $\mathcal{C}$: Set of classes
E: Number of epochs

function CLIENT_UPDATE(θ_k, ϕ_k, $\mathcal{C}$)
  for $t = 1, 2, \cdots, E$ **do**
    // With ϕ_k fixed
    Update θ_k by descending $\nabla_{\theta_k}\mathcal{L}_k$ ▷ Eq. 5
    // With θ_k fixed
    Update $\{\mathcal{S}_k^c\}_{c\in\mathcal{C}}$ ▷ Eq. 6
    Update ϕ_k ▷ Eq. 3
  return θ, ϕ

of the prediction head for class m in node j by the end of the t^{th} round. For the prediction head of class m in the PS, we aggregate the weights of prediction heads $\{\phi_{k,m}\}_k$ by: $\phi_{0,m} = \sum_k A_{k,m}\phi_{k,m}$. In contrast to

3) Training Strategy: Note, the prototypes ϕ is dependent on θ and ϕ should be fixed when performing prototype-based classification. Thus, it is unstable to minimize Eq. , we aggregate the model weights $\{\theta_k\}_k$ by :

$$\theta_0 = \sum_k \frac{\sum_m A_{k,m}}{\sum_{m,k} A_{k,m}}\theta_k.$$

Eq. reflects the *principle of maximum entropy* as $\frac{\sum_m A_{k,m}}{\sum_{m,k} A_{k,m}} = \frac{\sum_m A_{k,m}/C}{\sum_k \sum_m A_{k,m}/C}$, where each class is assigned with equal weights $\frac{1}{C}$.

Notably, Eq. is described for general cases. In practice, the calculation of Eq. can be significantly simplified if the binary prediction heads share parameters (if using a shared MLP as the binary prediction heads, $\|\phi_{k,m}^t - \phi_{0,m}^{t-1}\|_2^2$ shares the same value $\forall m$. An overview of the second stage is 5 by updating θ and ϕ simultaneously. In each client, the training is conducted in an alternative fashion, illustrated in Algorithm 21.

4) Theoretical Analysis on Divergence-Aware Aggregation Module: As a comparison, let us take a closer look at FedAVG, which is known for robustness against non-IID data. Let ϕ_0 be the model weights stored in the PS and $\{\phi_k\}_{k=1}^K$ be the model weights stored in each data node to be aggregated. For class j, we have $\phi_{0,j} = \sum_{k=1}^K \frac{n_k}{n}\phi_{k,j}$. Assume that node $m \neq j$ does not have any examples for class j (caused by class imbalance), the generated positive pseudo labels ("1" in Eq.) will be *noisy labels* [**?**]. Note $m \neq j$ implies that node m does not have ground truth for class j. In this case, $\phi_{m,j}$ will have decreased performance due to noisy information and $\frac{n_k}{n}$ can not properly reflect the weight of $\phi_{m,j}$ (say n_m is large)However, at the early phase of federated partially supervised training, as each client only has partially labeled data, $\mathcal{S}_k^c$ will be empty $\forall c \notin \mathcal{C}_k$ and μ_k^c is undefined. Thus, FedAvg might not be an optimal solution for FPSL.

After discussing the limitation of FedAVG, we analyze the training dynamics of the divergence-aware aggregation module (DAAM). First, we assume that the weight divergences for all data nodes are at the same level, there is no *outlier* (either too large or too small) among the data nodes. At the beginning of training, for class m, node m will have more labels than node $j \neq m$ which only have limited pseudo labels due to uncertainty.Thus, $n_{j,m}$ should be smaller than $n_{m,m}$ and there is a warm-up training phase where each client first trains with local ground truth partial labels in the supervised fashion. By replacing $\{\mathcal{S}_k^c\}_{c\in\mathcal{C}}$ with $\{\mathcal{S}_k^c\}_{c\in\mathcal{C}_k}$, we can acquire the local training procedure in the warm-up phase. Meanwhile, $\{\mathcal{S}_k^c\}_{c\in\mathcal{C}_k}$ can only generate $\{\mu_k^c\}_{c\in\mathcal{C}_k}$ by Eq. 3, *i.e.* $\phi_{m,m}^t$ should be assigned a larger weight than $\phi_{j,m}^t$. An extreme case is $n_{j,m} = 0$, where the weight for $\phi_{j,m}^t$ is 0, no contribution. When the federated training proceeds, more pseudo labels will be generated in node $j \neq m$ and a larger weight will be assigned to $\phi_{j,m}^t$ accordingly. By neglecting the weight divergences, there are missing prototypes in each client. However, Eq. 8 requires all prototypes for $\{\mu_k^c\}_{c\in\mathcal{C}}$ from all clients. Thus, in the warm-up phase, we simply aggregate global prototypes with Eq. simplifies to an instance of FedAvg by considering the effective labels.2 and Eq. 7, where the undefined prototypes will be automatically zeroed-out by the empty set, *i.e.* $|\mathcal{S}_k^c| = 0$.

If the weight divergences are significantly different, the reciprocal of the weight divergence ensures that the model with large weight divergence receives a small weight. Note, the large weight divergence is usually an undesired signal in FL [**?**]. In our case, after the warm-up training , the prediction heads trained with ground truth labels in a supervised fashion should gradually converge with small weight divergences. A large weight divergence indicates that the model is biased towards a class (or node). This is undesirable as the CR module could generate more pseudo labels for this class (or node), which further worsens the situation. The completed training scheme for federated partially supervised training is illustrated in Algorithm 2.

VI. EXPERIMENTS

The purposes of the conducted experiments are twofoldthreefold. 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 [**?**] have no findings of thoracic diseases), which makes federated partially supervised MLC even more difficult.

Algorithm 2 Training scheme for FedPSL.

$\mathcal{S}_k$: Partially labeled data in client k
$\mathcal{U}_k$: Unlabeled data in client k
$\theta_{0,0}$: Pre-trained model weights
$\theta_{k,t}$: Model weights of client k at the end of round t
$\phi_{k,0} = \{\mu_k^c\}_{c \in \mathcal{C}_k}$: Initial prototypes of client k extracted by $\theta_{0,0}$
$\phi_{k,t} = \{\mu_k^c\}_{c \in \mathcal{C}}$: Prototypes of client k at the end of round t
T_w: Number of warm-up training rounds
T: Number of training rounds

for $t = 1, 2, \cdots, T_w$ **do** ▷ Warm-up
  for $k = 1, 2, \cdots, K$ **do**
    $\theta_{k,t} \leftarrow \theta_{0,t-1}$ ▷ Synchronize with PS
    $\theta_{k,t}$, $\phi_{k,t} \leftarrow$ Client_Update($\theta_{k,t}$, $\phi_{k,t}$, $\mathcal{C}_k$)
  Upload $\{\theta_{k,t}\}_{k=1}^K$ and $\{\phi_{k,T_w}\}_{k=1}^K$ to PS to get $\theta_{0,t}$ and ϕ_{0,T_w} ▷ Aggregate with Eq. 2 and Eq. 7
for $t = T_w + 1, T_w + 2, \cdots, T$ **do**
  for $k = 1, 2, \cdots, K$ **do**
    $\theta_{k,t}, \phi_{k,t} \leftarrow \theta_{0,t}, \phi_{0,t}$ ▷ Synchronize with PS
    $\theta_{k,t}$, $\phi_{k,t} \leftarrow$ Client_Update($\theta_{k,t}$, $\phi_{k,t}$, $\mathcal{C}$)
  Upload $\{\theta_{k,t}\}_{k=1}^K$ to PS to get $\theta_{0,t}$ ▷ Aggregate with Eq. 2 and Eq. 7
  Upload $\{\phi_{k,t}\}_{k=1}^K$ to PS to get $\phi_{0,t}$ ▷ Aggregate with Eq. 2 and Eq. 8

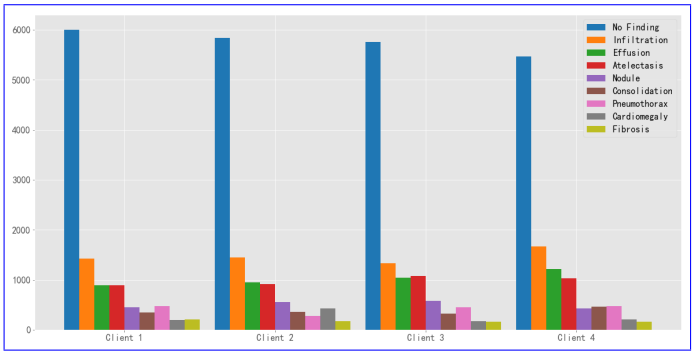

Fig. 3: Label statistics of four partially labeled training sets in four clients (*i.e.* $\mathcal{S}_1 - \mathcal{S}_4$)~~and~~ . "No Finding" means that a ~~fully labeled test set ($\mathcal{T}$) in two experiments~~CXR contains none of eight diseases of interest.

A. Data Preparation

We use the public dataset ChestX-ray14[3] [**?**] provided by the *National Institute of Health* (NIH). To facilitate reproducibility ~~, we use the~~ and preserve the non-IID data distribution of the original dataset, we use the default data ~~partitions of NIH. We utilize the original label distributions to reflect the label scarcity and class imbalance of non-IID datasets, as illustrated in Fig. 3. We use the first batch as the test set $\mathcal{T}$, which contains 4999 CXRs. We use the next four batches to create four data nodes and each data node contains 10000 CXRs~~partitions of NIH.

[3]https://nihcc.app.box.com/v/ChestXray-NIHCC/

B. Experimental Setup

Given the available computing resources, we set ~~$C = K = 4$~~ $K = 4$, *i.e.* there are four clients in our experiments. We ~~first create four partially labeled data nodes following Sec. **??**. We make two sets of thoracic diseases for two sets of experiments. The first set contains~~ use the four batches (batch 2 - batch 5) of ChestX-ray14 to create four clients (*i.e.* K = 4) where each client contains 10000 CXRs. We select eight most common thoracic diseases of interest (*i.e.* $C = 8$), which are *infiltration*, *effusion*, *atelectasis*, ~~and~~ *nodule*, ~~and the second setcontains *emphysema*, *edema*,~~ *consolidation*, *pneumothorax*, *cardiomegaly*, and *fibrosis*. The original label distributions of 8 diseases and as illustrated in Fig. 3. We also sample 500 positive cases and 500 negative cases for each of eight classes from other batches as an independent balanced test set.

To simulate the label scarcity and class imbalance across clients, we synthesize the partially labeled datasets by following two steps below. First, we only keep the labels of ~~*pneumonia*,~~ *infiltration* and *effusion* in the first client, the labels of *atelectasis* and ~~*hernia, which are the four most common and four least common diseases in the dataset, respectively. For each set of diseases*~~*nodule* in the second client, the labels of *consolidation* and *pneumothorax* in the third client, the labels of *cardiomegaly* and *fibrosis* in the fourth client. Second, we only keep the ~~corresponding partial labels in each data node, respectively~~labels of the first 2000 CXRs in each client and leave the rest 8000 CXRs unlabeled. For example, we only keep the labels of *infiltration* and ~~discard the remaining three diseases in data node~~ *effusion* in the first client and discard the remaining six diseases in $\mathcal{S}_1$ ~~for the first set of experiments and we only keep the labels of *hernia* in data node $\mathcal{S}_4$ in the second set of experiments~~.

The goal is to leverage four partially labeled datasets ($\mathcal{S}_1$ - $\mathcal{S}_4$) stored in $K = 4$ separated ~~data nodes~~ clients to learn a multi-label image classifier for ~~$C = 4$~~ $C = 8$ diseases. Note, due to data regulations, only model weights and metadata are allowed to be exchanged between the PS and each data node.

~~As the class distributions are extremely imbalanced, we consider two evaluation metrics in this study following [**?**], [**?**] . The first one is the F1-score, which is the harmonic mean of precision and recall. Without any prior knowledge, we set the threshold as 0.5 for all methods for a fair comparison. Note, in practice, the thresholds for different classes could be selected empirically, which is beyond the scope of this study. The second one is~~ We follow [**?**], [**?**] and choose *area under receiver operating characteristic* (AUROC) as the evaluation metric in this 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.

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 ~~three~~ six robust FPSL baselines~~, denoted as the first group (G1)~~.

- FedAVG is an adaptation of FedAVG [**?**] 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 ~~back-propagation~~backpropagation. Note, this is equivalent to standard SL~~as we only minimize Eq. ~~. In the PS, only the shared weights are aggregated and synchronized.
- FedProx is an adaptation of FedProx[4] [**?**] 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[5] [**?**] to a federated environment. The prediction heads are fine-tuned based on partial labels, in a fashion similar to FedAVG.
- FedCR is a federated adaptation of FixMatch[6] [**?**], a robust *semi*-SL method based on CR. As there is no existing PSL method designed for the problem of interest yet. We adapt FixMatch as a strong PSL baseline. For a fair comparison, we use the same set of hyperparameters (including warm-up training) as FedPSL in CR training.
- FedCR+ follows the same setup of FedCR, except the feature extractor is pre-trained by the same way of FedSSP.

The second group ~~(G2) includes four non-FPSL baselines, which are four methods with~~ includes three centralized baselines, where the corresponding constraints in Sec. ~~??~~ IV are relaxed (*i.e.* centralized training is feasible or full labels are available). We include these ~~four~~ two baseline PSL methods to provide an empirical understanding of the ~~negative~~ impact of FPSL and and supervised *Oracle* with full labels.

- ~~M1 is standard SL with centralized data and partial labels, where all partially labeled datasets are trained in a centralized setting. This is the centralized training counterpart of FedAVG~~IML [**?**] is the simplest PSL method, which simply ignores missing labels, *i.e.* only backpropagating the gradients corresponding to the partial labels.
- ~~M2 is FixMatch with centralized data and partial labels~~FixMatch is a centralized *semi*-SL method, which is adapted to PSL for the same reason of FedCR. We use the same set of hyperparameters of FedCR. This is also the centralized training counterpart of FedCR.
- ~~M3 is standard SL with decentralized data and full labels~~MixUp-PME [**?**] is centralized PSL method based on data augmentation and pseudo labeling. We use ~~FedAVG for federated training. This is the fully-labeled counterpart of~~FedAVG~~. In this study, we use M3 as the upper bound performance for decentralized data~~the default hyperparameters of [**?**].
- ~~M4~~ *Oracle* is standard SL with centralized data and full labels. This should be the best performance the model of interest can achieve under standard SL, which is also considered as the upper bound performance for centralized data.

[4]https://github.com/litian96/FedProx
[5]https://github.com/facebookresearch/simsiam
[6]https://github.com/google-research/fixmatch

D. Implementation

1) Data Pre-Processing: Each CXR has a resolution of 1024×1024. In the training process, each CXR is first resized to ~~256×256, and then randomly cropped to~~ 224×224. The image is 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 ~~256×256~~ 224×224 image, and μ and σ are the mean and standard deviation of the pixels of x. In the testing process, each CXR is also resized to 224×224, followed by instance normalization. We use the same data augmentation policy proposed in [**?**] for all methods in the training process.

2) Network Architecture: All baseline methods use a ~~ResNet18 [**?**]~~ DenseNet121 [**?**] as the encoder f_θ. We choose ~~ResNet18 as~~ DenseNet121 following [**?**] and it is a commonly adopted model in FL [**?**] for a lightweight experimental setup[7]. Each of the federated methods has $K+1$ ~~ResNet18s~~ DenseNet121s for K ~~nodes~~ clients and the PS, while each of the non-federated methods has one ~~ResNet18~~DenseNet121. All models are implemented in PyTorch on an NVIDIA Tesla V100. For the unsupervised pre-training stage in Sec. V-A, ~~the projector is a 3-layer MLP and the predictor is a 2-layer MLP. The feature dimension of the projector is 512 and the hidden dimension of the predictor is 128. Each fully-connected layer in the MLPs is followed by batch normalization [**?**] and ReLU [**?**], except the last layer. In the second stage, each prediction head is a fully-connected layer with hidden dimension 512.~~ we simply use SimSiam as it allows for a lightweight implementation.

3) Training: For a fair comparison, all networks are initialized with the same random seeds. ~~Note, to provide a comprehensive understanding of the problem, we do not use any additional labeled or unlabeled datasets in the training, as these could lead to impartial conclusions (domain shift should be taken into consideration if the pre-trained weights come from a different domain).~~ We train all methods ~~for 300 epochs. For methods including unsupervised pre-training, it consists of~~ with partial or full labels for 100 epochs~~of pre-training and 200 epochs of fine-tuning~~. The synchronization and aggregation for federated methods are performed every ~~$T=10$~~ 10 epochs. We use a standard Adam [**?**] optimizer with a fixed learning rate 10^{-3} for supervised training or partially supervised training. The binary cross-entropy in Eq. (5) 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

[7]The experiments with more advanced architectures are considered out of the scope of this work.

labeled data. For ~~the~~ methods including unsupervised pre-training~~stage~~, there are 200 epochs of pre-training with the federated version of SimSiam [?]. Following [?], we use a standard stochastic gradient descent optimizer with momentum of 0.9 ~~. The~~ and the initial learning rate is 0.05 with cosine annealing~~, following [?]. For CR-based methods, the~~ . The warm-up period is $T_w = 30$ for `FedPSL`, `FedCR`, `FedCR+`, and FixMatch.

VII. Conclusion

In this paper, we formulate and discuss a new problem ~~FPSL~~ federated partially supervised learning (FPSL) for decentralized partially labeled medical images. We also present `FedPSL`, a simple yet robust solution to FPSL. We are the first to leverage *self*-SL to mitigate label scarcity and class imbalance in FPSL. We further propose a ~~CR-based~~ local training module to mitigate label scarcity and a ~~divergence-aware~~ prototype-based global aggregation module to ~~make up for the cross-site class imbalance~~avoid large weight divergence. 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] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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

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

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

[22] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *NIPS*, vol. 34, 2017, pp. 4080–4090.

[23] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Pires, Z. Guo, M. Azar *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," in *NIPS*, vol. 33, 2020, pp. 21 271–21 284.

[24] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," *NIPS*, vol. 29, pp. 1163–1171, 2016.

[25] 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.

[26] 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.

[27] 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.

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

[29] 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.

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

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

[32] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.

[33] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9729–9738.

[34] C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, "Debiased contrastive learning," in *NIPS*, vol. 33, 2020, pp. 8765–8775.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[36] E. T. Jaynes, "Information theory and statistical mechanics," *Physical Review*, vol. 106, no. 4, p. 620, 1957.

[37] 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.

[38] H. Wang, W. Liu, A. Bocchieri, and Y. Li, "Can multi-label classification networks know what they don't know?" in *NIPS*, vol. 34, 2021.

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

[40] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *NIPS*, 2013, pp. 1196–1204.

[41] 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.

[42] 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.

[43] 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.

[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*. PMLR, 2015, pp. 448–456.

[45] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

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