

# Learning Underrepresented Classes from Decentralized Partially Labeled Medical Images

Anonymous Author

Anonymous Organization

\*\*\*@\*\*\*.\*\*\*

**Abstract.** Using decentralized data for federated training is one promising emerging research direction for alleviating data scarcity in the medical domain. However, in contrast to large-scale fully labeled data commonly seen in general object recognition tasks, the local medical datasets are more likely to only have images annotated for a subset of classes of interest due to high annotation costs. In this paper, we consider a practical yet under-explored problem, where underrepresented classes only have few labeled instances available and only exist in a few clients of the federated system. We show that standard federated learning approaches fail to learn robust multi-label classifiers with extreme class imbalance and address it by proposing a novel federated learning framework, FedFew. FedFew consists of three stages, where the first stage leverages federated self-supervised learning to learn *class-agnostic* representations. In the second stage, the decentralized partially labeled data are exploited to learn an energy-based multi-label classifier for the common classes. Finally, the underrepresented classes are detected with the learned energy and a *prototype*-based nearest-neighbor model is proposed for few-shot matching. We evaluate FedFew on multi-label thoracic disease classification tasks and demonstrate that it outperforms the federated baselines by a large margin.<sup>1</sup>

**Keywords:** Federated Learning · Partially Supervised Learning · Multi-Label Classification.

## 1 Introduction

Learning from partially labeled data, or partially supervised learning (PSL), has become an emerging research direction in label-efficient learning on medical images [25, 6, 18]. Due to high data collection and annotation costs, PSL utilizes multiple available partially labeled datasets when fully labeled data are difficult to acquire. Here, a partially labeled dataset refers to a dataset with only a specific *true subset* of classes of interest annotated. For example, considering a multi-label thoracic disease task on chest X-ray (CXR) images (*i.e.* a CXR could contain several diseases at the same time), a pneumonia dataset may only have labels for pneumonia but the labels for the other diseases of interest are missing.

---

<sup>1</sup> Code will be made available upon acceptance of the manuscript.

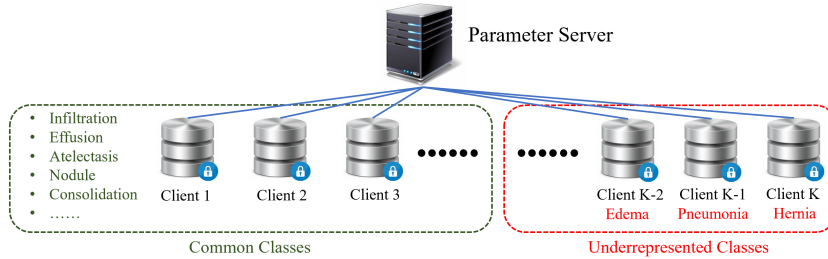


Fig. 1: An example problem setup with multi-label thoracic disease classification. In this example, each client in the red dashed box has only one underrepresented class, while the clients in the green dashed box share a set of common classes.

In this work, we extend the discussion of PSL to an unexplored federated setup, where the partially labeled datasets are stored separately in different clients (*e.g.* hospitals and research institutes). In this work, we denote the common classes (CCs) as the classes with enough labels to learn a multi-label classifier. Meanwhile, in a *open-world* scenario, there are also newly-found or under-examined classes, which tend to have much fewer labeled instances than the CCs. This extreme class imbalance makes learning these underrepresented classes (UCs) a difficult task. Furthermore, the main assumption of this work is that the CCs and UCs are annotated at disjoint clients, which makes this practical problem even more challenging. While a formal problem setup will be described in Sec. 2, an intuitive illustration of the problem is presented in Fig. 1.

**Contributions** We formalize this under-explored problem and present FedFew, the first solution to it. FedFew is a three-stage federated learning (FL) framework. Firstly, we utilize federated self-supervised learning (FSSL) to learn *class-agnostic* transferable representations in a pre-training stage. Secondly, in the fine-tuning stage, we propose an energy-based multi-label classifier that aims to utilize the knowledge from the CCs to learn representations for the UCs. Finally, we utilize a *prototype*-based nearest-neighbor classifier for few-shot matching given only few partially labeled examples for the UCs. We evaluate FedFew on a set of multi-label thoracic disease classification tasks on the Chest-Xray14 dataset [23] in a simulated federated environment. The empirical results show that the proposed framework outperforms existing methods by a large margin.

## 2 Problem Setup

**Task and Data Setup** The task of interest is multi-label classification (MLC) with  $C$  non-mutually exclusive classes of interest, where the decentralized data are stored in a federated system with  $K > 1$  clients. Let  $\mathcal{D}_k$  denote the data stored in client  $k \leq K$ , we have  $\mathcal{D}_k \cap \mathcal{D}_l = \emptyset$  for  $k \neq l$  and  $\{\mathcal{D}_k\}_{k=1}^K$  are non-IID data. For convenience, we define  $\mathcal{D}_k = \mathcal{P}_k \cup \mathcal{U}_k$ , where  $\mathcal{P}_k$  is a *partially labeled*

dataset and  $\mathcal{U}_k$  is an *unlabeled* dataset. We define  $n_k^p = |\mathcal{P}_k|$  and  $n_k^u = |\mathcal{U}_k|$ .<sup>2</sup> We assume that the classes of interest, denoted as  $\mathcal{C}$ , can be split into two mutually exclusive subsets, namely a set of UCs (denoted as  $\mathcal{C}_u \subset \mathcal{C}$ ), which is also the primary target of this work, and a set of CCs (denoted as  $\mathcal{C}_c = \mathcal{C} \setminus \mathcal{C}_u$ ). For simplicity, we consider a representative case that there are  $|\mathcal{C}_u| < K$  clients and each of these clients is annotated for only one UC<sup>3</sup>. For the remaining  $K - |\mathcal{C}_u|$  clients, we assume that each client has *partially labeled*<sup>4</sup> data for all CCs  $\mathcal{C}_c$ . We additionally require  $n_i^p \ll n_j^p \forall i \in \mathcal{C}_u, j \in \mathcal{C}_c$  to enforce the assumption of UCs. The learning outcome is to leverage the decentralized training data to train an MLC model for  $\mathcal{C}$ .

**Federated Environment Setup** In addition to  $K$  clients, there is a parameter server (PS) [14] for model aggregation. Let  $f_\theta$  be the model of interest. In the PS, the parameter set  $\theta_0$  is randomly initialized and sent out to  $K$  clients as  $K$  copies  $\{\theta_k\}_{k=1}^K$  for full synchronization. During the federated optimization phase, the client  $k$  updates  $\theta_k$  by training on  $\mathcal{D}_k$  independently for a number of local epochs. Then, the PS aggregates  $\{\theta_k\}_{k=1}^K$  collected from  $K$  clients to update  $\theta_0$ . Under the data regulations in the medical domain [20, 5], we assume that the patients' data (either raw data or encoded data) in a client can not be uploaded to the PS or other clients, *i.e.* only parameters  $\{\theta_k\}_{k=0}^K$  and *metadata* (*e.g.* the statistics of data) can be exchanged between the PS and the clients.

### 3 Method

In this section, we first provide the preliminaries that FedFew builds on in Sec. 3.1. The first training stage of FSSL is briefly described in Sec. 3.2, while the second training stage of energy-based federated learning with partial labels is described in Sec. 3.3. Finally, in Sec. 3.4, we present the prototype-based nearest-neighbor classifier for few-shot matching.

#### 3.1 Preliminaries

**FedAvg** As a seminal FL model, FedAvg [16] aggregates the model weights  $\{\theta_k\}_{k=1}^K$  as a weighted average. Mathematically, we have

$$\theta_0 = \sum_{k=1}^K a_k \theta_k, \quad (1)$$

<sup>2</sup>  $|\cdot|$  is the cardinality of a set.

<sup>3</sup> This is the most fundamental case. As a trivial extension, each client could have labels for more than one class and multiple clients could have labels for the same set of classes. The proposed method could be easily adapted to these extensions.

<sup>4</sup> Here, the images with CCs are partially labeled with respect to the missing labels of the UCs, *i.e.* they are fully labeled if we only consider CCs. The assumption here is that the CCs are diseases with high prevalence, which can be easily collected and diagnosed; but the UCs are rare and only spotted in certain clients.

where  $a_k = \frac{n_k}{n_{tot}}$ . The metadata  $n_k$  is the number of labeled training examples stored in client  $k$  and  $n_{tot} = \sum_{k=1}^K n_k$  is the total number of training examples. **Energy Function** Given a discriminative neural network classifier  $f$ , the energy function  $E(x; f) : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}$  maps an image with shape  $H \times W$  to a scalar, which is also known as *Helmholtz free energy* [13]. The energy is defined as

$$E(x; f) = -\tau \log \int_y \exp \frac{f^y(x)}{\tau}, \quad (2)$$

where  $f^y(x)$  is the logit of the  $y^{\text{th}}$  class label and  $\tau$  is the temperature parameter.

### 3.2 Federated Self-Supervised Learning

The first training stage consists of FSSL, where a feature extractor  $f_\theta$  is pre-trained to learn class-agnostic representations. Theoretically, multiple existing self-supervised learning frameworks (*e.g.* [8, 1, 7]) could serve as the local backbone. In this work, however, due to its lightweight nature, we leverage SimSiam [2]. Let  $\theta_0^t$  denote the aggregated model weights in the PS at the end of the  $t^{\text{th}}$  training round. Thus, at the beginning of the  $t+1^{\text{th}}$  round, the model weights at client  $k$  should be synchronized to  $\theta_0^t$ . After the local updates of the  $t+1^{\text{th}}$  round, the local model weights of client  $k$  are now  $\theta_k^{t+1}$  and,  $\theta_0^{t+1}$  is computed by applying Eq. 1 on  $\{\theta_k^{t+1}\}_{k=1}^K$ .

### 3.3 Energy-Based Federated Learning with Partial Labels

For standard MLC, it is common to use a  $C$ -dimensional binary vector to encode the label information for a given input. When all binary entries are 0s, the input does not contain any class of interest. However, with limited partial labels of UCs, it is difficult to train an MLC model for  $\mathcal{C}$  or  $\mathcal{C}_u$  directly. Instead, we first train an MLC model for CCs  $\mathcal{C}_c$ . In contrast to previous studies [23, 17], we encode the label into a  $(C_c + 1)$ -dimensional vector, where  $C_c = |\mathcal{C}_c|$ . That is to say, we use an additional dimension (denoted as  $0^{\text{th}}$  class<sup>5</sup>) to specifically determine whether the patient contains any CCs. Note, this  $0^{\text{th}}$  class only reflects the information on CCs, as we have no label information for the UCs.

Without loss of generality, let us consider a client  $k \in \mathcal{K}_c$  with only  $\mathcal{C}_c$  labeled, where  $\mathcal{K}_c$  denotes the clients with the CCs. Given an example  $x$  in client  $k$  with corresponding partial label  $y$ , the binary cross-entropy loss is

$$\mathcal{L}_{BCE}(x, y) = - \sum_{j \in \{0\} \cup \mathcal{C}_c} y_j \log(f_\theta^j(x)) + (1 - y_j) \log(1 - f_\theta^j(x)), \quad (3)$$

where  $f_\theta^j(x)$  is the probability score for the  $j^{\text{th}}$  class. As MLC can be decomposed into multiple binary classification tasks, the energy of  $x$  for class  $j$  degenerates

<sup>5</sup> When the additional dimension is 0, the rest of  $C_c$  dimensions should have at least one 1; when the additional dimension is 1,  $C_c$  dimensions should all be 0s.

**Algorithm 1** Energy-Based Federated Partially Supervised Training.  $T$  is the total number of rounds. We use  $\theta_k^t$  to denote the model weights stored in client  $k$  at the  $t^{\text{th}}$  round.

---

**Input:**  $\theta_0^0, \{\mathcal{P}_k\}_{k=1}^K, T_w, T$   
**Output:**  $\theta_0^T$

---

```

1: for  $t = 1, 2, \dots, T_w$  do                                ▷ Warm up
2:   for  $k \in \mathcal{K}_c$  do
3:      $\theta_k^t \leftarrow \theta_0^{t-1}$                                 ▷ Synchronize with PS
4:      $\theta_k^t \leftarrow \text{local\_update}(\theta_k^t)$                 ▷ Eq. (6)
5:    $\theta_0^t \leftarrow \sum_{k \in \mathcal{K}_c} a_k^t \theta_k^t$                 ▷ Aggregate with Eq. (1)
6: for  $t = T_w + 1, T_w + 2, \dots, T$  do
7:   for  $k = 1, 2, \dots, K$  do
8:      $\theta_k^t \leftarrow \theta_0^{t-1}$                                 ▷ Synchronize with PS
9:      $\theta_k^t \leftarrow \text{local\_update}(\theta_k^t)$                 ▷ Eq. (6) or Eq. (7)
10:   $\theta_0^t \leftarrow \sum_{k=1}^K a_k^t \theta_k^t$                 ▷ Aggregate with Eq. (1)
    
```

---

to  $E(x; f_\theta^j) = -\tau \log(1 + \exp^{\frac{f_\theta^j(x)}{\tau}})$  (c.f. Eq. (2)) and the *joint energy* [22] of  $x$  is then the sum of energies over all CCs  $\mathcal{C}_c$ :

$$E(x, f_\theta) = \sum_{j \in \{0\} \cup \mathcal{C}_c} E(x; f_\theta^j) = - \sum_{j \in \{0\} \cup \mathcal{C}_c} \tau \log(1 + \exp^{\frac{f_\theta^j(x)}{\tau}}). \quad (4)$$

We include a regularization term [15] to penalize the energy of  $x$  with a squared hinge loss:

$$\mathcal{L}_{E_c}(x) = \lambda \|\max(0, E(x; f_\theta) - m_c)\|_2^2, \quad (5)$$

where the margin  $m_c$  is a hyperparameter chosen empirically to decrease the energy of  $x$  and  $\lambda$  is a weight hyperparameter. The final optimization goal for client  $k \in \mathcal{C}_c$  is to minimize the sum of the two losses:

$$\mathcal{L}_c = \mathcal{L}_{BCE}(x, y) + \mathcal{L}_{E_c}(x). \quad (6)$$

For a client with an UC  $c_r \in \mathcal{C}_u$ , we only minimize a regularization term:

$$\mathcal{L}_u = \lambda \|\max(0, m_u - E(x; f_\theta))\|_2^2, \quad (7)$$

where the margin  $m_u$  is chosen empirically to increase the energy of  $x$ . Note, Eq. (5) and Eq. (7) are both designed to enlarge the *energy gap* between  $\mathcal{C}_c$  and  $\mathcal{C}_u$ . We aggregate the models weights  $\{\theta_k\}_{k=1}^K$  via Eq. (1), where  $a_k = \frac{n_k^p}{\sum_{j=1}^K n_j^p}$ . The complete pseudo-code is given in Algorithm 1.

### 3.4 Prototype-Based Inference

After the federated training in Sec. 3.3,  $f_\theta$  can be directly used as an MLC model to predict CCs.<sup>6</sup> Now, we use the energy (Eq. (4)) to detect UCs, *i.e.* if the

<sup>6</sup> Given the  $(C_c + 1)$ -dimensional output vector, we drop the 0<sup>th</sup> dimension and only use the  $C_c$ -dimensional vector as the final prediction.

energy of an example is lower than the threshold<sup>7</sup>, then the example is deemed to contain no UCs; if the energy is higher than the threshold, we further match the test example to the nearest neighbor, given few partially labeled examples. However, due to the constraint of data regulations, the training data and the test data are stored in separated clients. Thus, similar to [4], we transfer the metadata of UCs to the PS. Here, the metadata is the mean of the extracted features. For class  $c \in \mathcal{C}_u$ , we have

$$\mu_c^{pos} = \frac{\sum_{i=1}^{n_c^{pos}} g_\theta(x_i^{pos})}{n_c^{pos}}, \mu_c^{neg} = \frac{\sum_{i=1}^{n_c^{neg}} g_\theta(x_i^{neg})}{n_c^{neg}}, \quad (8)$$

where we use *pos* and *neg* to denote the positive and negative examples of class  $c$ , respectively, and use  $g_\theta$  to denote the feature extractor. Note, Eq. 8 factually defines the *prototype* in the few-shot learning literature [19]. With the *dual*-prototypes for class  $c$ , we match the test example to the closer one by computing the distance between the features of test example and the two prototypes<sup>8</sup>.

## 4 Experiments

### 4.1 Experimental Setup

To provide empirical insights into the problem of interest and ensure a fair comparison with the baselines, we share the same experimental setup (*e.g.* hyperparameters and dataset splits) among all experiments.

**Implementation** We explore two network backbones ( $f_\theta$ ), ResNet34 [9] and DenseNet121 [10], which are lightweight models commonly used for federated learning. We use SimSiam [7], a state-of-the-art SSL framework, to pre-train  $f_\theta$  locally, and use a standard Adam [12] optimizer with fixed learning rate  $10^{-3}$  and batch-size 64 for both pre-training and fine-tuning. We have  $\lambda = 0.1$ ,  $m_c = -10$ , and  $m_u = 10$ . We follow the same data pre-processing and augmentation procedure as [4]. The synchronization and aggregation for federated methods are performed every 10 epochs. For the second stage,  $T_w = 50$  and  $T = 100$ . All models are implemented in PyTorch (1.10.1) on an NVIDIA Tesla V100.

**Data** We use the multi-label dataset ChestX-ray14<sup>9</sup> [23] and adopt its default batch splits to ensure reproducibility. Based on the label statistics of the dataset, we choose *edema*, *pneumonia*, and *hernia* as the three UCs and use the remaining 11 classes as CCs. Note, most CXR images do not contain any diseases. We use 6 batches<sup>10</sup> to simulate the  $K = 6$  clients, where we use the first three batches to simulate the partially labeled datasets for CCs, where we randomly sample

<sup>7</sup> The threshold is chosen empirically to maximize the number of correctly classified training examples.

<sup>8</sup> Again, this is a simple case. When there are more than two prototypes collected from different clients for class  $c$ , majority voting is adopted.

<sup>9</sup> <https://nihcc.app.box.com/v/ChestXray-NIHCC>

<sup>10</sup> We use batch 2 to 7 in this work where each batch has  $10^4$  CXR images and similar label distributions.

Epoch	$r = 1$		$r = 0.5$		$r = 0.1$	
	RN	DN	RN	DN	RN	DN
100	61.69	70.47	69.34	71.94	65.92	73.81
200	62.96	71.22	68.32	71.36	65.24	74.43

Table 1: Impact of class imbalance on FSSL. We report the mean accuracy over three random seeds.

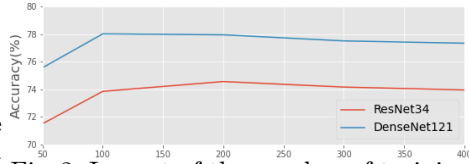


Fig. 2: Impact of the number of training epochs on FSSL.

$n_k^p = 5 \times 10^3$  images to keep partial labels for CCs. Each of the remaining three batches contains one of the three UCs. We sample 10 negative examples and 10 positive examples to simulate the class imbalance for UCs, *i.e.*  $n_k^p = 20$ . See Fig. 1 for an illustration of the class assignment among clients. From the remaining batches, we hold out 100 positive examples and 100 negative examples for each UC as the test set.

## 4.2 Results

**Empirical Analysis of FSSL** Following previous SSL studies, we examine the FSSL performance via the *linear classification protocol* [8, 1, 7]. Similar to [4], we fix all the weights of  $f_\theta$  except the last layer and only fine-tune the last layer on a public pneumonia dataset<sup>11</sup> [11]. In this dataset, there are three mutually exclusive classes, *normal*, *bacteria pneumonia*, and *virus pneumonia*. We randomly split the images of each class into two halves as the training and test sets. We use the test accuracy as the *proxy* measure to assess the representation learning performance. Firstly, we provide a counter-intuitive observation that more disease images might not improve the performance. We create two clients with Chest-Xray14 data, where one client contains  $10^4$  images without any diseases and the other client contains  $r \times 10^4$  images with various diseases. The results in Table 1 show that FedAvg does not always benefit from large  $r$  and it might be unnecessary to collect a large number of images with related diseases for pre-training. Secondly, we examine the impact of the training epochs for federated SSL in Fig. 2, where we pre-train  $f_\theta$  on the 6 clients described in Sec. 4.1. In contrast to the empirical findings collected from general images [8, 1, 7, 2], more epochs will not lead to diminishing performance gain but decreasing results. We will use DenseNet121 as the default network and use the pre-trained weights with 100 epochs in the following experiments.

**Evaluation of FedFew** Following the experimental setup in Sec. 4.1, we evaluate FedFew against a few seminal baselines from two aspects, *i.e.* we aim to achieve high accuracy on UCs while maintaining robust performance on CCs. The first baseline is a standard MLC model [17], which is trained with FedAvg on decentralized partially labeled data of all 14 classes and weighted binary cross-entropy [17]. We use *MLC w/o FSSL* and *MLC w/ FSSL* to differentiate whether  $f_\theta$  is pre-trained with FSSL. The second baseline is a nearest-neighbor

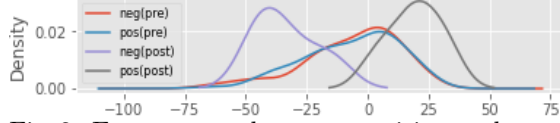
<sup>11</sup> <https://data.mendeley.com/datasets/rscbjbr9sj/2>

Method	Edema				Pneumonia				Hernia			
	A	P	R	F	A	P	R	F	A	P	R	F
MLC w/o FSSL	0.50	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.50	0.00	0.00	0.00
MLC w/ FSSL	0.50	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.50	0.00	0.00	0.00
NN (FSSL)	0.53	0.53	0.50	0.52	0.51	0.51	0.46	0.48	0.50	0.50	0.54	0.52
NN (MLC w/ FSSL)	0.53	0.53	0.50	0.52	0.51	0.51	0.46	0.48	0.50	0.50	0.54	0.52
FedFew w/o EBM	0.71	0.85	0.50	0.63	0.69	0.84	0.46	0.59	0.72	0.83	0.54	0.65
FedFew w/ EBM	<b>0.75</b>	<b>1.00</b>	<b>0.50</b>	<b>0.67</b>	<b>0.73</b>	<b>1.00</b>	<b>0.46</b>	<b>0.63</b>	<b>0.77</b>	<b>1.00</b>	<b>0.54</b>	<b>0.70</b>

Table 2: Performance comparison on the UCs over three random seeds. The standard MLC and NN models fail to predict the UCs.

Method	AUROC
MLC w/o FSSL	0.59 $\pm$ 0.08
MLC w/ FSSL	0.61 $\pm$ 0.07
FedFew w/o EBM	0.64 $\pm$ 0.05
FedFew w/ EBM	<b>0.66 <math>\pm</math> 0.05</b>

Table 3: Performance comparison on the CCs over 3 active cases before (pre) and after (post) *EBM* training in Client 4.



classifier (*NN*), where  $g_\theta$  is either pre-trained with FSSL alone or further fine-tuned with *MLC w/ FSSL*. For FedFew, we use *EBM* to denote the energy-based loss in the training. The results on UCs are presented in Table 2 and include the mean accuracy (A), precision (P), recall (R), and F-1 (F) score over three random seeds. Note, standard MLC models fail to detect any images of the UCs due to extreme class imbalance. With only prototypes (as only metadata can be transferred to the PS), NNs struggle to improve over random guessing. FedFew (*w/o EBM*) outperforms the two baselines by a large margin while *EBM* further improves the performance of FedFew with higher precision. Similar to Table 2, we report the mean AUROC for the 11 CCs with standard deviation over three runs in Table 3, where FedFew achieves robust performance on CCs.

Distance	F-1
Euclidean	0.60
Earth Mover's	0.43
Cosine	<b>0.67</b>

Table 4: Comparison of distances.

As an ablation study, we visualize the energy density plots between images with and without UCs in Fig. 3, which demonstrates that including *EBM* in the training does increase the energy gap, thus leading to improved performance. We consider three distance metrics for Sec. 3.4, which are cosine distance [21], Euclidean distance [19], and earth mover's distance [24] (computed with Sinkhorn-Knopp algorithm [3]). We choose the cosine distance based on the empirical robustness shown in Table 4, where average F-1 scores over the UCs are reported.

## 5 Conclusion

In this work, we raise awareness of an under-explored problem, namely the learning of underrepresented classes from decentralized partially labeled medical images. We not only provide a solution to this novel problem but also provide the first empirical understanding of federated partially supervised learning with extreme class imbalance, a new research direction on label-efficient learning.



## References

1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607. PMLR (2020)
2. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR. pp. 15750–15758 (2021)
3. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: NIPS. vol. 26, pp. 2292–2300 (2013)
4. Dong, N., Voiculescu, I.: Federated contrastive learning for decentralized unlabeled medical images. In: MICCAI. pp. 378–387. Springer (2021)
5. European Commission: General data protection regulation (2016), [https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu\\_en](https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en)
6. Fang, X., Yan, P.: Multi-organ segmentation over partially labeled datasets with multi-scale feature abstraction. IEEE TMI (2020)
7. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Pires, B., Guo, Z., Azar, M., et al.: Bootstrap your own latent: A new approach to self-supervised learning. In: NIPS. vol. 33, pp. 21271–21284 (2020)
8. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR. pp. 9729–9738 (2020)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
10. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 4700–4708 (2017)
11. Kermany, D.S., Goldbaum, M., Cai, W., Valentim, C.C., Liang, H., Baxter, S.L., McKeown, A., Yang, G., Wu, X., Yan, F., et al.: Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell* **172**(5), 1122–1131 (2018)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
13. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. *Predicting Structured Data* **1**(0) (2006)
14. Li, M., Andersen, D.G., Smola, A.J., Yu, K.: Communication efficient distributed machine learning with the parameter server. In: NIPS. pp. 19–27 (2014)
15. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. In: NIPS. vol. 33, pp. 21464–21475 (2020)
16. McMahan, B., Moore, E., Ramage, D., Hampson, S., Aguera y Arcas, B.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS. pp. 1273–1282. PMLR (2017)
17. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al.: Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225* (2017)
18. Shi, G., Xiao, L., Chen, Y., Zhou, S.K.: Marginal loss and exclusion loss for partially supervised multi-organ segmentation. *Medical Image Analysis* p. 101979 (2021)
19. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: NIPS. pp. 4077–4087 (2017)
20. US Department of Health and Human Services: Health insurance portability and accountability act (2017), <https://www.cdc.gov/phlp/publications/topic/hipaa.html>
21. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: NIPS. pp. 3630–3638 (2016)

22. Wang, H., Liu, W., Bocchieri, A., Li, Y.: Can multi-label classification networks know what they don't know? In: NIPS. vol. 34 (2021)
23. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: CVPR. pp. 2097–2106 (2017)
24. Zhang, C., Cai, Y., Lin, G., Shen, C.: Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In: CVPR. pp. 12203–12213 (2020)
25. Zhou, Y., Li, Z., Bai, S., Wang, C., Chen, X., Han, M., Fishman, E., Yuille, A.L.: Prior-aware neural network for partially-supervised multi-organ segmentation. In: ICCV. pp. 10672–10681 (2019)