Learning with Partial Labels: An Exploratory Study of Data Scarcity for Medical Images



Nanqing Dong
Keble College
University of Oxford

A qualifying dissertation submitted for
Doctorate of Philosophy in Computer Science

Trinity 2023

To my beloved motherland, the meaning of my life

Statement of Originality

I hereby declare that this thesis is submitted to the Department of Computer Science, University of Oxford, in fulfillment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, it describes my own research.

I also declare that, for the publications associated with this thesis, I am the only contributor in terms of conceptualization, investigation, methodology, implementation, experimentation, visualization, and writing. The co-authors only provided necessary assistance in the form of funding acquisition, literature review, proofreading, and supervision.

August 5, 2023

Nanqing Dong

Abstract

The data-driven nature of deep learning models is often associated with the use of large-scale fully labeled datasets. However, the high cost of annotation often makes these datasets unavailable for standard supervised learning. This thesis explores an emerging learning paradigm called partially supervised learning (PSL), which deals with partially labeled examples. Unlike semi-supervised learning, where an example is either fully labeled or unlabeled, PSL addresses a similar but distinct problem. In PSL, each example is labeled for a subset of sub-tasks within the context of multi-task learning. The utilization of multiple small-scale partially labeled datasets in PSL is still a challenging research question. Therefore, in this thesis, I provide a formal definition of PSL for the first time.

Partially labeled datasets are frequently encountered in the medical field due to the high cost of annotation and the specific clinical objectives they serve. In this regard, I leverage partially supervised multi-label classification (PSMLC) on medical images as the task of interest to exemplify the aforementioned concept. This thesis aims to offer empirical insights into enhancing the performance of deep learning models trained on small-scale partially labeled datasets by presenting the following methodological advancements under four difference PSL setups.

This thesis first proposes a practical solution to PSL under data scarcity based on vicinal risk minimization (VRM). While VRM was originally designed as a simple technique to enhance the generalization ability of deep learning models in supervised learning, I demonstrate that, by leveraging principle of maximum entropy, VRM can be successfully applied to PSL. Through experimental evidence, I illustrate that VRM shows advantages over semi-supervised learning methods in both accuracy and computational complexity, particularly in practical and challenging scenarios. This thesis represents the first application of VRM to PSL, offering both

theoretical insights and empirical findings. Furthermore, I provide an explanation for the limitations of semi-supervised learning methods in this context.

Fueled by the recent renaissance of self-supervised learning, this thesis aims to explore the potential of utilizing unlabeled data to enhance the performance of deep learning models trained on partially labeled data. While self-supervised learning has played a crucial role in pre-training deep learning models for fully-supervised downstream tasks, its impact on downstream tasks involving only partially labeled data remains unclear. Therefore, this thesis presents the first empirical study to investigate the contributions of self-supervised learning in addressing PSL problems and proposes a novel pretext task based on VRM.

While previous studies have focused on situations where partially labeled datasets are collected in a centralized fashion or can be remotely accessed without communication constraints, this thesis takes a step further to discuss situations where partially labeled datasets are decentralized under data governance constraints. Specifically, the partially labeled datasets are stored separately in different clients within a federated system, and the raw data cannot be exchanged between the clients or uploaded to the central server. This motivates the definition of a new research problem called federated partially supervised learning (FPSL). In this thesis, I theoretically discuss the challenges of FPSL and provide a robust federated solution to address this problem.

Moreover, I observe that in practical scenarios, classes often follow a long-tailed distribution, which means that, in addition to common classes, there are also rare classes. Consequently, I discuss a special case of FPSL known as federated partially supervised learning with underrepresented classes. In this case, a few clients store a limited number of partially labeled examples related to the rare classes, while the other clients have a much larger number of partially labeled examples related to common classes. To address this extreme class imbalance, I propose, for the first time, an energy-based federated solution to effectively learn and recognize the rare classes without compromising the performance of common classes.

In summary, this thesis explores four unexplored research directions of PSL under data scarcity. For each of four research directions, a novel

solution is proposed to address the problem of interest and evaluated with simulated experiments. The empirical results not only provide empirical insights to understand PSL but also pose new research directions on PSL.

Acknowledgements

I must acknowledge the selfless help from my departmental supervisor, Prof. Irina Voiculescu. Without her support, especially during the pandemic, I cannot finish my DPhil degree. Her kindness, patience, and tolerance inspire me and unleash my potential to conduct research in the areas that I am interested in at the University of Oxford. I hope our technical contributions can make a difference in the machine learning community in the long run. Words cannot express my utmost gratitude to her, who decided to accept me as her DPhil student and made every fantastic thing happen at Oxford.

I would like to thank the Department of Computer Science, University of Oxford, for accepting me into its prestigious DPhil program and generously providing the *Department of Computer Science Scholarship* to support my DPhil study. The department offers me not only an unparalleled research platform that facilitates this thesis but also numerous academic and research opportunities that significantly shape my academic career. As a DPhil student, I have successfully developed my skills in research, teaching, communication, and management.

I would like to thank Keble College, University of Oxford, for accepting me as its matriculated student and enriching my student life with various fantastic college and MCR activities. Keble College provides me with the cheapest college housing and also the closest room to the Wolfson Building of the Department of Computer Science throughout my DPhil study, which not only alleviates my financial burden but also keeps my time cost of commuting between my college room and my departmental office within 4 minutes for a round trip. As the recipient of the *Graduate Support Fund* multiple times, I want to show my gratitude to Keble College for its generous funding, which supports the conference registration for my conference presentations.

I would like to thank Amazon, Google, and Huawei for providing free cloud computing services to support my research. The empirical studies in this thesis heavily rely on the GPUs included in the cloud computing service. Without their free cloud computing service, the experiments in this thesis cannot be finished on time.

I would like to thank the National Institute of Health, United States of America, for providing the public benchmark dataset used in this thesis.

I would like to thank my parents for their love and support during my DPhil study. The period of my DPhil study significantly overlaps with the pandemic, a long period of difficult time for everyone. Their encouragement not only helped me make up my mind to pursue a doctoral degree but also provided the strongest backing during this difficult time. Without them, I might not have had the courage to finish this thesis.

I would like to thank my grandparents. My grandfather (my father's father) is a national hero who passed away even before I was born. But his story motivates me to become a real man who can contribute to the country and human society. My grandmother (my father's mother), who raised me since I was a child, passed away before my confirmation exam. I promised her that I would earn my doctoral degree. This promise also kept me working hard during this difficult time. My mother's parents also raised me, and their kindness also influenced my life. Therefore, this thesis also serves as a result of their upbringing in my childhood.

I would like to thank my country, which provides a safe living environment and solid fundamental education. The culture of my country enlightens me to develop a strong heart and never give up.

The greatest failure in the world is the failure to try.

Contents

1 Introduction **1**

1.1 Motivations 2
1.2 Research Questions 3
1.3 Contributions 3
1.4 Thesis Outline 4
1.5 Publications 5

2 Partially Supervised Learning **7**

2.1 Preliminaries 7
2.1.1 Multi-Task Learning 7
2.1.2 Semi-Supervised Learning 8
2.1.3 Multi-Label Classification 9
2.2 Problem Formulation 10
2.2.1 Definition 10
2.2.2 Challenges of Partially Supervised Learning 12
2.2.2.1 Data Scarcity 13
2.2.2.2 Class Imbalance 13
2.2.2.3 Domain Shift 14
2.2.2.4 Partial Supervision 14
2.3 Partially Supervised Multi-Label Classification 15
2.3.1 Task Setup 15
2.3.2 Evaluation Metrics 16
2.3.2.1 F1 Score 16
2.3.2.2 AUROC 17
2.3.2.3 Choice of Evaluation Metric 18
2.4 Literature Review 18
2.4.1 Partially Supervised Learning in Medical Domain 18
2.4.2 Limitations of Existing Studies 19

2.5 Summary . . . 20

3 Partially Supervised Learning with Vicinal Risk Minimization 21

3.1 Introduction . . . 21
3.2 Related Work . . . 23
3.2.1 Vicinal Risk Minimization . . . 23
3.2.2 Partially Supervised Multi-Label Classification . . . 23
3.3 Preliminaries . . . 25
3.3.1 Vicinal Risk Minimization . . . 25
3.3.2 MixUp . . . 26
3.3.3 Weighted Loss . . . 26
3.4 Challenges of Vicinal Risk Minimization for Partially Supervised Multi-Label Classification . . . 27
3.4.1 Challenge of Generating Vicinal Images . . . 27
3.4.2 Challenge of Generating Vicinal Labels . . . 28
3.5 Method . . . 28
3.5.1 Locally Full Supervision . . . 28
3.5.2 Globally Partial Supervision . . . 28
3.5.3 Training Strategy . . . 29
3.6 Theoretical Analysis . . . 32
3.6.1 Theoretical Guarantee . . . 32
3.6.2 Enlarged Sample Space . . . 33
3.6.3 Computational Cost . . . 33
3.6.4 Relationship with Noisy Labels . . . 33
3.7 Experiments . . . 34
3.7.1 Experimental Setup . . . 34
3.7.1.1 Data . . . 34
3.7.1.2 Models . . . 35
3.7.1.3 Implementation . . . 35
3.7.2 Empirical Analysis . . . 36
3.7.2.1 Comparison with Baselines . . . 36
3.7.2.2 Robustness Under Class Imbalance . . . 38
3.7.2.3 Sensitivity to α . . . 38
3.7.2.4 Adaptive MixUp-PME . . . 38
3.7.2.5 Robustness under Multi-Task Learning . . . 43
3.8 Conclusion . . . 44

4 Partially Supervised Learning with Self-Supervised Pre-Training 45
4.1 Introduction . . . 45
4.2 Related Work . . . 49
4.2.1 Pretext Tasks . . . 49
4.2.2 Contrastive Learning . . . 50
4.3 Preliminaries . . . 52
4.3.1 Rotation Prediction . . . 52
4.3.2 Masked Image Modeling . . . 53
4.3.3 Instance Discrimination . . . 53
4.4 Task Affinity in Transfer Learning . . . 54
4.4.1 Task Affinity . . . 54
4.4.2 Proxy Evaluation . . . 55
4.4.3 Hypotheses of Proxy Task Affinity . . . 57
4.5 Multi-Pretext-Task Learning . . . 58
4.6 Reverse Vicinal Risk Minimization . . . 58
4.7 Experimental Setup . . . 60
4.7.1 Data . . . 60
4.7.2 Implementation . . . 60
4.7.2.1 Rotation Prediction . . . 61
4.7.2.2 Masked Image Modeling . . . 61
4.7.2.3 Instance Discrimination . . . 61
4.7.2.4 Data Augmentation . . . 61
4.7.2.5 Proxy Task Affinity . . . 62
4.7.2.6 Multi-Pretext-Task Learning . . . 63
4.7.2.7 Vicinal Risk Minimization . . . 63
4.8 Empirical Analysis . . . 63
4.8.1 Relationship between Pretext and Downstream Tasks . . . 63
4.8.2 Impact of Self-Supervised Pre-Training . . . 65
4.8.3 Effect of Multi-Pretext-Task Learning . . . 70
4.8.4 Evaluation of Reverse Vicinal Risk Minimization . . . 71
4.9 Conclusion . . . 75

5 Federated Partially Supervised Learning 76
5.1 Introduction . . . 76
5.2 Related Work . . . 81
5.2.1 Federated Learning . . . 81

5.2.2 Partially Supervised Learning under Decentralization 83
5.2.3 Data Governance 83
5.3 Problem Setup 84
5.4 Towards Understanding Federated Partially Supervised Learning . . . 85
5.4.1 A Multi-Task Representation of Multi-Label Classification . . 85
5.4.2 A Closer Look at FedAVG 86
5.4.3 Challenges of Federated Partially Supervised Learning 86
5.4.3.1 Impact on Task-Specific Predictor 87
5.4.3.2 Impact on Task-Agnostic Feature Extractor 88
5.5 Method 88
5.5.1 Task-Dependent Model Aggregation 88
5.5.2 Task-Agnostic Decoupling Learning 89
5.6 Experiments 90
5.6.1 Datasets 91
5.6.1.1 ChestX-ray14 Dataset 91
5.6.1.2 Tuberculosis Chest X-ray Database 92
5.6.1.3 COVID-19 Detection Dataset 92
5.6.2 Experimental Setup 92
5.6.3 Baselines 93
5.6.4 Implementation 94
5.6.4.1 Data Pre-Processing 94
5.6.4.2 Network Architecture 95
5.6.4.3 Training 95
5.6.5 Federated System without One-Class Clients 95
5.6.5.1 Data Setup 95
5.6.5.2 Impact of Partial Supervision 96
5.6.5.3 Analysis of Centralized PSL Methods 96
5.6.5.4 Impact of Decentralization 96
5.6.5.5 Comparison with Federated Baselines 97
5.6.5.6 Analysis of `FedPSL` 98
5.6.6 Federated System with One-Class Clients 98
5.6.6.1 Data Setup 98
5.6.6.2 Impact of One-Class Clients 98
5.6.6.3 Analysis of `FedPSL` 100
5.6.7 Robustness Analysis 102
5.6.7.1 Sensitivity to Data Scarcity 102

5.6.7.2 Sensitivity to Extreme Clients . . . 102
5.6.7.3 Impact of Synchronization Frequency . . . 103
5.7 Conclusion . . . 103

6 Federated Partially Supervised Learning for Rare Classes 106
6.1 Introduction . . . 107
6.2 Related Work . . . 109
6.2.1 Few-Shot Learning . . . 109
6.2.2 Out-of-Distribution Detection . . . 110
6.3 Problem Setup . . . 111
6.4 Preliminaries . . . 113
6.4.1 Energy-Based Models . . . 113
6.4.1.1 Energy Function . . . 113
6.4.1.2 Connection with Machine Learning . . . 114
6.4.2 Distance Metrics . . . 114
6.4.2.1 Cosine Distance . . . 115
6.4.2.2 Squared Euclidean Distance . . . 115
6.4.2.3 Earth Mover's Distance . . . 115
6.5 Method . . . 117
6.5.1 Federated Self-Supervised Pre-Training . . . 117
6.5.1.1 Motivation . . . 117
6.5.1.2 Training . . . 117
6.5.2 Energy-Based Federated Learning with Partial Labels . . . 118
6.5.2.1 Motivation . . . 118
6.5.2.2 Training . . . 119
6.5.3 Class-Aware Inference . . . 121
6.5.3.1 Inference for Common Classes . . . 121
6.5.3.2 Inference for Rare Classes . . . 122
6.5.3.3 Remarks . . . 123
6.5.4 Theoretical Analysis . . . 123
6.6 Experiments . . . 124
6.6.1 Experimental Setup . . . 124
6.6.1.1 Data . . . 124
6.6.1.2 Simulation . . . 124
6.6.2 Implementation . . . 125
6.6.2.1 Data Pre-Processing . . . 125

6.6.2.2 Network Architecture . . . 125
6.6.2.3 Training . . . 125
6.6.2.4 Evaluation Metrics . . . 126
6.6.3 Empirical Analysis . . . 126
6.6.3.1 Impact of Federated Self-Supervised Pre-Training . . . 126
6.6.3.2 Evaluation of Energy-Based Federated Training . . . 127
6.6.3.3 Analysis on Energy . . . 128
6.6.3.4 Choice of Distance Metric . . . 130
6.7 Conclusion . . . 131

7 Conclusion **132**
7.1 Summary . . . 132
7.1.1 Contributions . . . 132
7.1.2 Limitations . . . 133
7.2 Future Directions . . . 133
7.3 Concluding Remarks . . . 135

Bibliography **136**

List of Figures

2.1 STL vs MTL. In contrast to STL, MTL utilizes shared hidden representations (in purple), instead of task-specific hidden representations. Here, the inputs (or the data modalities of the inputs) for different tasks are assumed to be identical. Thus, the task-specific encoders can also be replaced by a shared encoder (in purple). 8

2.2 Illustration of domain shift in medical imaging. Three chest X-ray images collected from three different hospitals exhibit clear visual difference in brightness and contrast. Three chest X-ray images are selected from Wingspan Dataset [38]. 15

3.1 Illustration of vicinal images generated by state-of-the-art VRM methods on two visually similar input chest X-ray images. (a) and (b) are input images, while (c) - (d) are vicinal images. The mixing ratio $\lambda = 0.75$ is used for all four methods displayed and the value is chosen for illustration purpose. See Section 3.3.2 for the definition of λ. . . . 24

3.2 Illustration of vicinal images generated by state-of-the-art VRM methods on two visually different input chest X-ray images. (a) and (b) are input images, while (c) - (d) are vicinal images. The mixing ratio $\lambda = 0.75$ is used for all four methods displayed and the value is chosen for illustration purpose. See Section 3.3.2 for the definition of λ. . . . 25

3.3 (a) and (b) are two golden retrievers from ImageNet [34]. (c) and (d) are two chest X-ray images from ChestX-ray14 [177]. In contrast to object-centric images, chest X-ray images have multiple objects of semantic interests (*e.g.* organs) within the same image. 26

3.4 Generating a vicinal image and a vicinal label by the proposed method. The input images are two chest X-ray images. There are two classes of interest, which are cardiomegaly and hernia. The binary label "1" denotes that the image contains the class of interest and "0" denotes that the image does not contain the class. For example, for the first image, $[1, 0]$ means that the image only contains the symptoms of cardiomegaly and does not have signs of hernia. λ is set as 0.7. When full labels are available for both images, Equation (3.2) can be directly applied to generate the vicinal labels for all classes of interest. Two possible situations are depicted when both images are partially labeled. "?" denotes the missing binary label. Given a specific class, as long as the first image has a ground truth label for it, Equation (3.5) can still generate a vicinal label for this class. 30
3.5 Sensitivity of MixUp-PME to α for different classes. MixUp-PME has different optimal α values for different classes. 40
3.6 Label statistics of positive cases for the partially labeled training set and fully labeled test set. 43

4.1 Pretext task of rotation prediction (clockwise). The network is trained to predict the rotation degree. 52
4.2 Pretext task of masked image modeling. The network is trained to reconstruct the masked image. 53
4.3 Pretext task of instance discrimination. The network is trained to learn the invariance between two augmented views of the same image. . . . 54
4.4 Illustration of the concept of task affinity in optimization. Intuitively, when three tasks denote three unit vectors in linear algebra, the dot product between any two tasks can denote a hypothetical task affinity score between them. Let Task 1 (the blue arrowed line tangent to the arc) be the anchor task (*i.e.* the downstream task). The ideal pretext task should optimize θ along with the direction of Task 1. Let the gray dashed line (perpendicular to Task 1) denote the zero task affinity score. Task 2 should have a positive task affinity score with Task 1 as the gradient is still in a direction that loss can be decreased. Task 3 should have a negative task affinity score with Task 1 as the gradient direction increases the loss. However, in practice, there is no such quantitative measurement. 56

4.5 Pretext task based on reverse vicinal risk minimization. Given two randomly sampled images and the generated vicinal images, the network is trained to predict the MixUp hyperparameter λ. 59
4.6 Learning curves for three different pretext tasks. The losses for all pretext tasks converge. 64
4.7 Linear classification performance for three different pretext tasks. The reported AUROCs are the mean mAUROCs over three random seeds. 65
4.8 Downstream task performance for different values of n, the number of partially labeled instances for each class given specific pre-training epochs. The reported AUROCs are the mean mAUROCs over three random seeds. 66
4.9 Downstream task performance with different numbers of pre-training epochs given specific numbers of partially labeled instances for each class (n). The reported AUROCs are the mean mAUROCs over three random seeds. 67
4.10 Linear classification performance of multi-pretext-task learning. The reported AUROCs are the mean mAUROCs over three random seeds. 71
4.11 Downstream task performance of multi-pretext-task learning with different numbers of pre-training epochs given specific numbers of partially labeled instances for each class (n). The reported AUROCs are the mean mAUROCs over three random seeds. 72
4.12 Learning curve for the pretext task based on vicinal risk minimization. 74
4.13 Linear classification performance of the pretext task based on vicinal risk minimization. The reported AUROCs are the mean mAUROCs over three random seeds. 74
4.14 Downstream task performance of the pretext task based on vicinal risk minimization with different number of pre-training epochs given n. The reported AUROCs are the mean mAUROCs over three random seeds. 75

5.1 Illustration of federated partially supervised learning for a multi-label classification task on chest X-ray images. Here, each client (data node) is annotated for only one thoracic disease. This simplified example is used to convey the main concept of the problem of interest in this chapter. In practice, each client could be partially labeled for more than one class. In the first client, a model can only utilize the label information of *infiltration* for an image but the model does not have access to the label information of the other three diseases assuming that there are only four diseases of interest in total. To ensure data governance, only model weights and the metadata (*e.g.* statistics) of the local data can be communicated between each client and the parameter server (see Section 5.3 for a formal description). In the current scenario, the goal of federated partially supervised learning is to utilize the four partially labeled datasets stored in the four different clients to train the model of interest in the parameter server. 78

5.2 Visual illustration of multi-site chest X-ray images. (a) is diagnosed with cardiomegaly and emphysema. (b) is tuberculosis positive. (c) is COVID-19 positive. 92

5.3 Sensitivity of the overall performance (AUROC) to the size of the local dataset (n). The overall performance is measured by the mean AUROC over $C = 6$ classes in $K = 4$ clients. 103

5.4 Impact of extreme clients on the overall performance (AUROC). n denotes the size of the local dataset in the extreme clients, while each common client has 1000 partially labeled examples. Two out of four clients are the extreme ones. The overall performance is measured by the mean AUROC over $C = 6$ classes in $K = 4$ clients. 104

6.1 Illustration of federated partially supervised learning with rare classes (extreme class imbalance) for a multi-label classification task on chest X-ray images. In this example problem setup, the clients (data nodes) in the green dashed box are annotated for the same set of common classes and each client in the red dashed box are only annotated for one underrepresented class. From the perspective of a federated system with K clients, each example is partially labeled, *i.e.* either partially labeled for common classes or partially labeled for a single underrepresented class. 112

6.2 Impact of the number of training epochs on federated self-supervised pre-training. More epochs will not lead to diminishing performance gain but decreasing results. 127
6.3 Energy gaps between positive and negative cases in clients with rare classes without *EBM* in federated training. Under a standard federated training procedure, the energy distributions of positive and negative cases significantly overlap. It is difficult to differentiate the positive cases from the negative cases. 130
6.4 Energy gaps between positive and negative cases in clients with rare classes with *EBM* in federated training. Under the proposed energy-based federated training procedure, the energy distributions of positive and negative cases show clear difference in contrast to Figure 6.3. . . 131

List of Tables

2.1 Comparison between semi-supervised learning and partially supervised learning. 13

3.1 A step-by-step demonstration of Listing 3.1. After getting $\tilde{y}$ with Equation (3.2) and/or Equation (3.5) with $\lambda = 0.75$, the binary cross entropy losses for $C = 4$ classes are computed following Equation (3.3). However, only the binary cross entropy losses of the first and the third classes will be back-propagated as the binary cross entropy losses of the second and the fourth classes are zeroed out. As a comparison, MixUp can only be applied to the first class, when both input labels are available. 31

3.2 $\tilde{y}^k$ generated by Equation (3.2) when both y_i^k and y_j^k are known. . . . 32

3.3 $\tilde{y}^k$ generated by Equation (3.5) when only y_i^k is known. 32

3.4 $\tilde{y}^k$ generated by Equation (3.5) when only y_i^k is known. 32

3.5 Comparison between learning-based pseudo labeling methods and the proposed method. 34

3.6 Comparison with baselines. In the training set, each chest X-ray image is only partially labeled for one class. The proposed MixUp-PME achieves the best overall performance. The reported numbers are F1 scores. Blue denotes the highest number in each column (other than the *Oracle*). 37

3.7 Comparison with baselines under intra-class positive-negative ratio $r = \frac{1}{9}$. In the training set, each chest X-ray image is only partially labeled for one class. The proposed MixUp-PME achieves the best overall performance. The reported numbers are F1 scores. Blue denotes the highest number in each column (other than the *Oracle*). 39

3.8 Highest performance that can be achieved by MixUp-PME for each class. The repThe last row "α" records the corresponding α value when the highest F1 scores can be achieved by MixUp-PME for each class. 40
3.9 AMP *vs*. MixUp-PME. In the training set, each chest X-ray image is only partially labeled for one class. AMP achieves competitive performance against MixUp-PME. The reported numbers are F1 scores. Blue denotes the highest number in each column (other than the *Oracle*) 41
3.10 Comparison between MixUp-PME and *vanilla* under multi-task learning. The reported numbers are F1 scores. Blue denotes the highest number in each column (other than the *Oracle*). 42

4.1 Class-wise downstream task performance. The reported numbers are the mean and standard deviation of AUROCs over three random seeds. Blue denotes the highest AUROC for each class (each row). 68
4.2 Class-wise downstream task performance for MPTL and `RVRM`. The reported numbers are the mean and standard deviation of AUROCs over three random seeds. Blue denotes the highest AUROC for each class among MPTL baselines (except the last column). Red denotes that `RVRM` outperforms the highest AUROC for each class in Table 4.1. 73

5.1 A subset of similarities and differences among FL, distributed learning, and edge computing. 82
5.2 Label statistics of positive cases for the experimental setup **without** one-class clients in the federated system. "0" denotes that the client does not have any partial labels for the disease, *i.e.* this disease is unlabeled. 96
5.3 Quantitative evaluation of multi-label thoracic disease classification in a federated system **without** one-class clients. The reported number are the mean and standard deviation of average AUROCs over three random seeds. Blue denotes the highest number in each column (other than the *Oracle*). The column "**Average**" denotes the average AUROCs over classes, which is considered as the overall performance for a particular method. 99
5.4 Label statistics of positive cases for the experimental setup **with** one-class clients in the federated system. "0" denotes that the client does not have any partial labels for the disease, *i.e.* this disease is unlabeled. 100

5.5 Quantitative evaluation of multi-label thoracic disease classification a federated system **with** one-class clients. The reported number are the mean and standard deviation of average AUROCs over three random seeds. Blue denotes the highest number in each column (other than the *Oracle*). The column "**Average**" denotes the average AUROCs over classes, which is considered as the overall performance for a particular method. 101
5.6 Impact of synchronization frequency on the overall performance (AUROC). The reported numbers are the mean and standard deviation of average AUROCs over three random seeds. The overall performance is measured by the mean AUROC over $C = 6$ classes. E denotes the number of local training epochs. Blue denotes the highest number in each column (other than the *Oracle*). 105

6.1 Impact of class imbalance on federated self-supervised pre-training. The mean accuracy over three random seeds are reported. For ResNet34, $r = 0.5$ gives the highest accuracy; for DenseNet121, $r = 0.1$ gives the highest accuracy. Overall, a large r (*e.g.* $r = 1$) might not help, *i.e.* it is unnecessary to collect a large number of images with diseases. . . . 127
6.2 Performance comparison on the rare classes. The standard MLC and NN models fail to predict the rare classes. A, P, R, and F denote the mean accuracy, precision, recall, and F1-score over three random seeds, respectively. Blue denotes the highest number in each column. 129
6.3 Performance comparison on the common classes. AUROC denotes the mean AUROC of eleven common classes over three random seeds. Following [140], only the mean AUROC is reported. Blue denotes the highest number. 130
6.4 Performance comparison on the choice of distance metrics. F1 denotes the mean F1-score of three rare classes over three random seeds. Blue denotes the highest number. The cosine distance leads to the highest F1-score. 131

Abbreviations

AUROC ..... area under the curve of the receiver operating characteristic

BCE ..... binary cross entropy

BLO ..... bi-level optimization

CT ..... computed tomography

CXR ..... chest X-ray

EBM ..... energy-based model

ERM ..... empirical risk minimization

FL ..... federated learning

FPSL ..... federated partially supervised learning

GPU ..... graphic processing unit

IID ..... independent and identically distributed

ML ..... machine learning

MLC ..... multi-label classification

MTL ..... multi-task learning

MRI ..... magnetic resonance imaging

OOD ..... out-of-distribution

PME ..... principle of maximum entropy

PSL ..... partially supervised learning

PSMLC ..... partially supervised multi-label classification

ROC ..... receiver operating characteristic

SGD ..... stochastic gradient descent

VRM ..... vicinal risk minimization

Chapter 1

Introduction

Fueled by advances in deep learning research, breakthroughs have been made in various fields of artificial intelligence, including computer vision [152], natural language processing [170], speech recognition [24], and robotics [123]. Deep learning models, as data-driven methods, demonstrate their true potential when large-scale fully labeled training data is available. A dataset is considered fully labeled when each instance in the dataset has complete annotations relevant to the tasks of interest. For instance, ImageNet [34] has served as a benchmark dataset for image classification, where human curators categorize each image into one of a thousand classes. With over 1.28 million object-centric images, ImageNet enables machine learning scientists to develop transferable representations for various downstream tasks such as object detection [143] and semantic segmentation [20].

However, acquiring such a "perfect" dataset with sufficient fully labeled data for supervised training can be prohibitively expensive. The data collection and curation process involves considerations such as data accessibility, storage, privacy, and the cost of human annotation, among others. This challenge becomes even more pronounced in the medical domain, where available medical datasets are relatively small, annotations require clinical expertise, and patient data is confidential and protected by privacy laws [53, 166]. In practice, there are more "imperfect" datasets in the wild. These datasets may be small-scale or have missing annotations for the tasks of interest. Rather than solely relying on the collection of large-scale fully labeled datasets, an alternative option is to leverage these imperfect datasets.

The objective of this thesis is to address an under-explored yet practical research question: how to effectively utilize datasets with missing annotations. Distinguished from previous studies on semi-supervised learning, this thesis focuses on an emerging learning paradigm called *learning from partial labels* or, in other words, *partially supervised learning*. The term "partial labels" is derived from the concept of "full

labels". In modern machine learning, a learner is often expected to simultaneously solve multiple tasks, which is referred to as multi-task learning [17]. In multi-task learning, "full labels" denotes the situation that an instance is annotated for all the tasks of interest and "partial labels" denotes the situation that an instance is only annotated for a few tasks of interest. For example, in multi-label classification, a learner aims to determine whether an instance belongs to each class of interest. Classifying an instance into a specific class can be considered a task, and that instance is considered partially labeled if only a few classes are annotated. Similarly, in dense prediction tasks like semantic segmentation, each pixel of an image is assigned to a semantic class. Classifying a pixel can be seen as a task, and the image is considered partially labeled if only a subset of pixels have annotations.

1.1 Motivations

As an exploratory study on learning from partial labels, this thesis has four primary motivations.

Firstly, the problem formulation of learning from partial labels, as a novel learning paradigm, lacks a formal definition. To address this gap, it is necessary to establish a standardized definition of the learning environment, learning outcomes, and key terminologies such as "partial label" and "partially labeled dataset". This standardization will facilitate future research in this field.

Secondly, there is currently no existing literature review available for the learning paradigm under investigation. Therefore, it would greatly benefit the machine learning community to provide a comprehensive summary of existing methods and identify current bottlenecks.

Thirdly, given that learning from partial labels is still in its early stages, there are limited application scenarios available. It is essential to explore more challenging problem setups beyond the existing straightforward ones.

Fourthly, there is a lack of empirical understanding of the problem, particularly in challenging setups. Therefore, it would be valuable to offer empirical insights to the machine learning community, further enhancing their understanding of this problem domain.

1.2 Research Questions

The primary research goal of this thesis is to understand PSL under data scarcity. As an exploratory work, this thesis studies partially supervised multi-label classification (PSMLC) under four unexplored data scarcity setups. I formulate and address the problems of PSL under these four different setups. These problems can be formulated as four primary research questions.

For the first research question, I aim to address PSMLC with only small-scale partially labeled data. The answer to this research question is presented in Chapter 3.

For the second research question, I aim to address PSMLC with additional unlabeled data. The key difference between the first research question and second research question is how to leverage the unlabeled data. The answer to this research question is presented in Chapter 4.

For the third research question, I aim to address PSMLC under a decentralized setup, where the partially labeled datasets are stored in different sites with communication constraints. The answer to this research question is presented in Chapter 5.

For the fourth research question, I aim to address PSMLC under a decentralized setup with additional extreme class imbalance. The key difference between the third research question and fourth research question is how to mitigate the cross-site class imbalance. The answer to this research question is presented in Chapter 6.

Before answering these four research questions in corresponding chapters, the necessary background knowledge and the definitions of main concepts throughout this thesis are provided in Chapter 2. This is also the first work that formally introduces PSL.

1.3 Contributions

Chapter 3 For the first time, the problem of partially supervised multi-label classification under data scarcity is investigated. A novel solution based on vicinal risk minimization is presented to augment both the input image space and the input label space simultaneously. Theoretical guarantees are achieved to ensure the quality of the augmented image-label pairs. The experimental results provide initial empirical insights into future research directions on partially supervised learning under data scarcity.

Chapter 4 First of all, the term task affinity is formally defined for label-efficient learning as a link to study the relationship between pre-training and fine-tuning. This chapter presents the first empirical study of the relationship between the pretext task and the downstream task in partially supervised multi-label classification, as well as the first empirical study of the impact of self-supervised pre-training in partially supervised multi-label classification. Moreover, this chapter serves as the first study of multi-pretext-task learning in self-supervised pre-training for partially supervised multi-label classification. A self-supervised task based on vicinal risk minimization is proposed for the first time and demonstrates robust performance in both representation learning and downstream partially supervised multi-label classification, competing favorably with state-of-the-art self-supervised learning methods.

Chapter 5 For the first time, the problem of federated partially supervised learning for decentralized medical images is formulated, and the challenges of federated partially supervised learning are explained. A novel federated partially supervised training pipeline for multi-label classification under data scarcity is proposed. The proposed solution includes a task-dependent model aggregation module and a task-agnostic decoupling learning module. Additionally, I provide initial evidence that federated partially supervised learning is an under-explored problem compared to existing learning paradigms, and I offer the community the first benchmark for federated partially supervised multi-label classification, accompanied by a set of performance evaluations and baseline comparisons.

Chapter 6 For the first time, the problem of federated partially supervised learning for underrepresented classes is formulated and discussed. A novel energy-based solution for federated partially supervised multi-label classification of rare classes is proposed. I provide an initial empirical understanding of the problem and demonstrate the practical difficulties of federated partially supervised learning for rare classes.

1.4 Thesis Outline

The remaining chapters of this thesis are structured as follows:

Chapter 2 Chapter 2 provides the necessary background on machine learning and formally defines the main concepts relevant to understanding the problem of interest. This chapter also summarizes the common data challenges faced by the problem and reviews the existing literature.

Chapter 3 Chapter 3 discusses partially supervised learning under data scarcity and proposes a novel data augmentation technique based on vicinal risk minimization [19] to mitigate data scarcity. The experiments provide empirical insights into partially supervised learning under data scarcity.

Chapter 4 Chapter 4 explores the potential of using self-supervised pre-training to improve the learning performance of partially supervised learning. This chapter also highlights the issue of *task misalignment* between the self-supervised task in pre-training and the partially supervised task. The discussion on task misalignment leads to two questions: how to quantify this task misalignment and how to choose an appropriate self-supervised task for the target partially supervised task. The chapter provides tentative answers to these questions and proposes a novel self-supervised task based on vicinal risk minimization, which can be linked to Chapter 3.

Chapter 5 Chapter 5 investigates a new research area of partially supervised learning, specifically in a federated setup. This research question is formulated as *federated partially supervised learning* in this thesis. The chapter studies the impact of data decentralization on partially supervised learning and the challenges associated with it. It also presents a novel federated framework based on meta-learning [78] to address the issues of data decentralization and partial supervision.

Chapter 6 Chapter 6 extends the discussion from Chapter 5 by focusing on federated partially supervised learning with extreme class imbalance, where rare or underrepresented classes exist and are partially labeled. This chapter examines the impact of extreme class imbalance on federated partially supervised learning and proposes a novel energy-based federated solution to address this challenging scenario.

Chapter 7 Chapter 7 serves as the concluding chapter of this thesis, summarizing the key contributions of each chapter. Additionally, it discusses potential future research directions in the field of partially supervised learning.

1.5 Publications

There are six publications associated with this thesis by the time this thesis is finished. I declare that, for the papers listed below, I am the first author, corresponding author, and sole contributor in terms of conceptualization, investigation, methodology, implementation, experiment, visualization, and writing. The co-authors only

provide necessary assistance in the form of funding acquisition, literature review, proofreading, and supervision.

The list below is shown in chronological order.

[40] Self-Supervised Multi-Task Representation Learning for Sequential Medical Images. Nanqing Dong, Michael Kampffmeyer, Irina Voiculescu. *European Conference on Machine Learning*, 2021.

[44] Federated Contrastive Learning for Decentralized Unlabeled Medical Images. Nanqing Dong, Irina Voiculescu. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2021.

[39] Towards Robust Partially Supervised Multi-Structure Medical Image Segmentation on Small-Scale Data. Nanqing Dong, Michael Kampffmeyer, Xiaodan Liang, Min Xu, Irina Voiculescu, Eric Xing. *Applied Soft Computing*. 2022.

[45] Revisiting Vicinal Risk Minimization for Partially Supervised Multi-Label Classification Under Data Scarcity. Nanqing Dong, Jiayi Wang, Irina Voiculescu. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2022.

[41] Learning Underrepresented Classes from Decentralized Partially Labeled Medical Images. Nanqing Dong, Michael Kampffmeyer, Irina Voiculescu. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2022.

[42] Federated Partially Supervised Learning for Decentralized Medical Images. Nanqing Dong, Michael Kampffmeyer, Irina Voiculescu, Eric Xing. *IEEE Transactions on Medical Imaging*, 2022.

Chapter 2 and Chapter 3 are both associated with [39, 45]. Chapter 4 is associated with [40, 44, 41]. Chapter 5 is associated with [44, 42]. Chapter 6 is associated with [41, 44].

Chapter 2

Partially Supervised Learning

This chapter defines formally the problem of interest of this thesis, namely partially supervised learning (PSL).

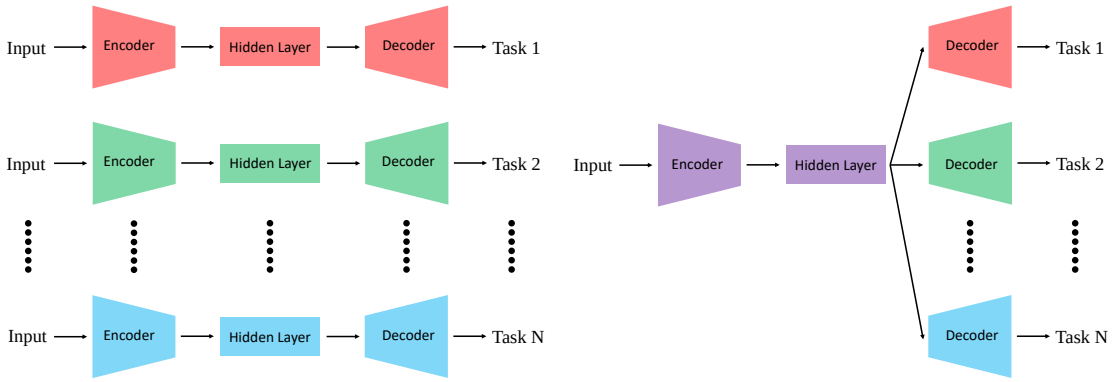
2.1 Preliminaries

2.1.1 Multi-Task Learning

A traditional methodology in machine learning is to learn tasks one by one [17], where large complex tasks are decomposed into small, simple, and reasonably independent sub-tasks. This learning paradigm is referred to as single-task learning (STL). In contrast to STL, multi-task learning (MTL) aims to improve generalization performance via an *inductive transfer* mechanism between tasks [17, 196]. Inductive transfer utilizes additional sources of information to improve the performance of learning on the task of interest. It has been shown [17] that inductive transfer can improve generalization (accuracy on unseen data) and convergence (the speed of learning).

The *inductive bias* (also known as learning bias) in machine learning is the set of assumptions that the learner uses to predict outputs of given unseen inputs [122]. From the perspective of inductive bias, MTL can also be interpreted as leveraging the extra training signals of related tasks as inductive bias to improve generalization of a specific task.

Let x be an input in the sample space $\mathcal{X}$ (*i.e.* $x \in \mathcal{X}$), y be the corresponding *target* of x in the label space (*i.e.* $y \in \mathcal{Y}$), and t be a task of interest in the task space $\mathcal{T}$ (*i.e.* $t \in \mathcal{T}$). In STL, each task is assumed to be independent. Assume the model of interest is a neural network and can be denoted as $g \circ f$, *i.e.* $(g \circ f)(x) = g(f(x))$, where f denotes the hidden layer representations and g denotes the last layer. Given a training set $\mathcal{S}_{train}$ and a test set $\mathcal{S}_{test}$, the learning outcome of training $g \circ f$ on



(a) STL (b) MTL

Figure 2.1: STL vs MTL. In contrast to STL, MTL utilizes shared hidden representations (in purple), instead of task-specific hidden representations. Here, the inputs (or the data modalities of the inputs) for different tasks are assumed to be identical. Thus, the task-specific encoders can also be replaced by a shared encoder (in purple).

$\mathcal{S}_{train}$ is to minimize the *empirical risk* on $\mathcal{S}_{test}$:

$$\sum_{(x,y)\in\mathcal{S}_{test}} \sum_{t\in\mathcal{T}} \mathcal{L}(g_t \circ f_t(x), y^t) \tag{2.1}$$

where $\mathcal{L}$ is the loss function, $g_t \circ f_t$ is the task-specific model for task t, and $y_t \in y$ is the target for task t.

MTL trains multiple tasks in parallel with shared hidden layer representations. Thus, representations f_t in Equation (2.1) are replaced with shared representations f_{shared}, and the empirical risk of MTL on $\mathcal{S}_{test}$ is now:

$$\sum_{(x,y)\in\mathcal{S}_{test}} \sum_{t\in\mathcal{T}} \mathcal{L}(g_t \circ f_{shared}(x), y^t) \tag{2.2}$$

The evolution from STL to MTL is illustrated in Figure 2.1.

2.1.2 Semi-Supervised Learning

A closely-related learning paradigm to PSL is semi-supervised learning [18, 163, 185]. In machine learning, supervised learning considers the situation that that training data are fully labeled and unsupervised learning considers the situation where training data are unlabeled. Semi-supervised learning is a special case that falls between supervised learning and unsupervised learning, where the training dataset contains both labeled data and unlabeled data.

Let $(x_l, y_l) \in \mathcal{X} \times \mathcal{Y}$ be an input-label pair and $x_u \in \mathcal{X}$ be an unlabeled input. For semi-supervised learning, the training set $\mathcal{S}_{train}$ can be split into two non-overlapping sub-sets, including a full labeled one $\mathcal{S}_L$ and an unlabeled one $\mathcal{S}_U$, *i.e.* $\mathcal{S}_{train} = \mathcal{S}_L \cup \mathcal{S}_U$. It is commonly to assume $\mathcal{S}_U$ is significantly larger than $\mathcal{S}_L$, *i.e.* $\mathcal{S}_L \ll \mathcal{S}_U$.

It is worth mentioning that semi-supervised learning can refer to either *transductive learning* [58] or *inductive learning* [119]. The goal of transductive learning is to assign the correct labels to the given unlabeled data $\mathcal{S}_U$ and the goal of inductive learning is to utilize both $\mathcal{S}_L$ and $\mathcal{S}_U$ to learn a correct mapping $\mathcal{F}$ from $\mathcal{X}$ to $\mathcal{Y}$, *i.e.* $\mathcal{F} : \mathcal{X} \mapsto \mathcal{Y}$. To compare partially supervised learning and semi-supervised learning under a similar problem setup, the scope of semi-supervised learning is constrained to be inductive learning in this thesis.

2.1.3 Multi-Label Classification

A representative instance task of MTL is multi-label classification (MLC) [164, 112]. MLC is a generalization of binary classification from simple task to multiple tasks. In MLC, a learner is expected to recognize multiple classes at the same time. In contrast to *multi-class classification*, another generalization of binary classification from simple task to multiple tasks, the sub-tasks (classes) in MLC are independent (non-mutually exclusive), *i.e.* an input can be categorized as multiple classes at the same time (*c.f.* an input can only belong to one class for multi-class classification).

Assume there are C classes of interest for MLC and the set of classes of interest can be denoted as $\mathcal{C}$, *i.e.* $C = |\mathcal{C}|$. Following the definitions in Section 2.1.1, the goal of binary classification is to learn a mapping $g_c \circ f_c : \mathcal{X} \mapsto \{0, 1\}$ for class $c \in \mathcal{C}$. Thus, the goal of MLC is to learn a mapping:

$$\boldsymbol{g} \circ f_{shared} : \mathcal{X} \mapsto \{0, 1\}^C \tag{2.3}$$

where $\boldsymbol{g} = \{g_c\}_{c \in \mathcal{C}}$.

For a binary classification task, binary cross entropy (BCE) loss [64] is defined as:

$$\mathcal{L}_{BCE}(x, y) = -y \log p(y = 1|x) - (1 - y) \log p(y = 0|x) \tag{2.4}$$

where $p(y = 1|x)$ is the predicted probability of x being class status 1 and y refers to a single task here. Then, the loss function for MLC is:

$$\mathcal{L}_{MLC}(x, \mathbf{y}) = \sum_{c=1}^{C} \mathcal{L}_{BCE}(g_c \circ f_{shared}(x), y^c) \tag{2.5}$$

where $y^c \in \mathbf{y}$ is a binary label for class c and $\mathbf{y}$ is commonly represented as a C-element binary vector $[y^1, y^2, \cdots, y^C]^T$.

While there are various MTL tasks (*e.g.* multi-class semantic segmentation [60] and multi-class object detection [198]), MLC is chosen as the main task in this thesis to illustrate the concept of PSL and to evaluate the proposed contributions for two reasons:

- MLC has a simple mathematical formulation. This can benefit theoretical analysis.
- MLC requires moderately low computational resource. This ensures scalability of experimental design.

2.2 Problem Formulation

2.2.1 Definition

The first step is to clearly define the main concepts that are necessary to understand the problem of interest. As there is no formal definition presented in the existing studies, I make the following definitions to describe the partially labeled data on my own. These definitions are the key to understand the research questions presented in Section 1.2 and corresponding data scarcity setups in Chapter 3 – Chapter 6.

Let (x, y) be an input-label pair. A *label* y is considered *complete* if complete label information regarding the task of interest t is available for the input x. In this case, the label is also denoted as *full label* in the literature [39]. Analogous to full label, the key concept of PSL is *partial label*, *i.e.* only incomplete label information is available for x. In contrast to a full label, a partial label has missing label information regarding the same input x. The partial label can have completely different definitions in different contexts. Taking image as an example, the partial label can refer to situation where only part of pixels are labeled for the downstream task and the rest of pixels are unlabeled. In this thesis, the partial label is defined under the perspective of MTL.

Definition 2.2.1 (Partial Label)**.** An instance is partially labeled if and only if all the following requirements are met.

1. There are multiple tasks of interest $\mathcal{T}$ or the task of interest can be decomposed into multiple sub-tasks $\mathcal{T}$, *i.e.* $|\mathcal{T}| > 1$.
2. Only a *true* subset of $\mathcal{T}$ is annotated, denoted as $\mathcal{T}_L$, *i.e.* $\mathcal{T}_L \subset \mathcal{T}$ or $|\mathcal{T}_L| < |\mathcal{T}|$.

3. $\mathcal{T}_L$ is non-empty, *i.e.* $\mathcal{T}_L \neq \emptyset$ or $|\mathcal{T}_L| > 0$.

With Definition 2.2.1, it is easy to define a *partially labeled dataset*. In a general sense, a dataset is partially labeled if at least one example in the dataset is partially labeled. However, this could lead to ambiguity when describing the dataset. Thus, the following definition is adopted throughout the thesis when mentioning a partially labeled dataset.

Definition 2.2.2 (Partially Labeled Dataset)**.** A dataset $\mathcal{D}$ is partially labeled with respect to a set of tasks of interest $\mathcal{T}$ if and only if all the following requirements are met.

1. $\mathcal{D}$ is non-empty, *i.e.* $\mathcal{D} \neq \emptyset$ or $|\mathcal{D}| > 0$.
2. The dataset can be represented in the form of $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, where x is an example and y is its corresponding partial label following Definition 2.2.1.
3. $\forall x \in \mathcal{D}$, x is partially labeled for the same subset of tasks $\mathcal{T}_L$.

Definition 2.2.2 is conditional on $\mathcal{T}$ when $\mathcal{D}$ is a fully labeled dataset with respect to $\mathcal{T}_L$. If $\mathcal{T}_L = \mathcal{T}$, Definition 2.2.2 becomes the definition of a fully labeled dataset. Similarly, if $\mathcal{T}_L = \emptyset$, Definition 2.2.2 can also be used to describe an unlabeled dataset, where y is now a null value.

A direct result of Definition 2.2.2, the training set $\mathcal{S}_{train}$ can be split into multiple subsets with respect to a set of tasks of interest $\mathcal{T}$. Here, a general definition is provided.

Definition 2.2.3 (Partially Labeled Training Set – General)**.** A training set $\mathcal{S}_{train}$ is partially labeled with respect to a set of tasks of interest $\mathcal{T}$ if and only if all the following requirements are met.

1. $\mathcal{S}_{train}$ can be split into $K > 1$ multiple non-overlapping non-empty subsets, *i.e.* $\mathcal{S}_{train} = \bigcup_{i=1}^{K} \mathcal{D}_i$.
2. A subset $\mathcal{D}_i$ is annotated with the set of tasks $\mathcal{T}_i$ and $\mathcal{D}_i$ can be fully labeled ($\mathcal{T}_i = \mathcal{T}$), partially labeled ($\mathcal{T}_i \subset \mathcal{T}$), or unlabeled ($\mathcal{T}_i = \emptyset$) with respect to $\mathcal{T}$.
3. At least one subset $\mathcal{D}_i$ in $\mathcal{S}_{train}$ is partially labeled according to Definition 2.2.2.
4. Each task has at least one annotated example, *i.e.* $\mathcal{T} = \bigcup_{i=1}^{K} \mathcal{T}_i$.

Definition 2.2.3 describes a general case, where $\mathcal{S}_{train}$ can also contain fully labeled data and unlabeled data. As one of goals in this thesis is to provide an empirical analysis on the impact of partial supervision, it is more meaningful to focus on partially labeled data only. A special definition is provided below. I use "general" and "special" to differentiate two different setups.[1]

Definition 2.2.4 (Partially Labeled Training Set – Special)**.** A training set $\mathcal{S}_{train}$ is partially labeled with respect to a set of tasks of interest $\mathcal{T}$ if and only if all the following requirements are met.

1. $\mathcal{S}_{train}$ can be split into $K > 1$ multiple non-overlapping non-empty subsets, *i.e.* $\mathcal{S}_{train} = \bigcup_{i=1}^{K} \mathcal{D}_i$.
2. Each subset $\mathcal{D}_i$ in $\mathcal{S}_{train}$ with corresponding annotated set of tasks $\mathcal{T}_i$ is partially labeled according to Definition 2.2.2.
3. Each task has at least one annotated example, *i.e.* $\mathcal{T} = \bigcup_{i=1}^{K} \mathcal{T}_i$.

With all ingredients defined above, the formal definition of PSL is presented below.

Definition 2.2.5 (Partially Supervised Learning)**.** Partially supervised learning is a field of multi-task learning devoted to understanding and building methods that leverage a partially labeled training set to improve performance on some set of tasks.

From Definition 2.2.5, PSL can be viewed as a subfield of MTL. Thus, the learning outcome of MTL, formulated in Equation (2.2), also applies to PSL.

2.2.2 Challenges of Partially Supervised Learning

As an exploratory study on an emerging research area, it is worthwhile discussing the challenges of PSL. In the first step, I review semi-supervised learning as PSL and semi-supervised learning share similarities on problem formulation and methods. In contrast to PSL, semi-supervised learning has been studied over decades [168]. A direct comparison between PSL and semi-supervised learning is first made in Table 2.1 to provide an initial insight. The major differences between them can be summarized as below.

[1]The terms "general" and "special" are motivated by the *general theory of relativity* and *special theory relativity* proposed by Albert Einstein.

1. For semi-supervised learning, each instance is either fully labeled or unlabeled. For PSL, each sub-task is either fully labeled or unlabeled for each instance.

2. For semi-supervised learning, labeled data and unlabeled data commonly come from the same or two different sources. In other words, $\mathcal{S}_L$ and $\mathcal{S}_U$ can be easily separated from $\mathcal{S}_{train}$. For PSL, partially labeled data might come from multiple sources and each partially labeled dataset is annotated for a different subset of tasks.

3. The goal of semi-supervised learning is to use unlabeled instances to perform inductive transfer and the goal of PSL is use unlabeled tasks to perform inductive transfer.

Table 2.1: Comparison between semi-supervised learning and partially supervised learning.

	Semi-Supervised Learning	Partially Supervised Learning
Supervision	instance-wise	task-wise
Data Sources	one or two domains	multiple domains
Goal	use unlabeled instances to perform inductive transfer	use unlabeled tasks to perform inductive transfer

To this end, four major challenges are listed below.

2.2.2.1 Data Scarcity

In this thesis, the terms "data scarcity" and "label scarcity" are used interchangeably to describe the situations that only limited *labeled* data are available in the training set, either instance-wise or task-wise. As discussed in Section 2.1.2, in semi-supervised learning, it is common to assume that the number of unlabeled data significantly outperforms the number of labeled data. Similarly, PSL faces the same challenge of data scarcity. Thus, PSL is also a label-efficient learning paradigm to maximally utilize the available labeled information.

2.2.2.2 Class Imbalance

Similar to semi-supervised learning, the second challenge is task imbalance or class imbalance if each class is considered as a sub-task. For convenience, I use the class imbalance for consistency with the literature [86]. Most of the time, the term "class imbalance" refers to the *inter-class imbalance*, where certain classes have many more

labeled examples than the rest of classes. In a practical scenario, the real distribution for the population is commonly *non-independent and identically distributed* (non-IID), *e.g.* a long-tailed distribution. Additionally, for MLC, the exemplar MTL task of this thesis, the term "class imbalance" also refers to *intra-class imbalance*. That means, for each class, the ratio between the positive and negative cases is small, *i.e.* the binary distribution severely imbalanced. This is common in the medical domain, where there are more negative cases (patients without certain disease) than positive cases (patients with certain disease) in the population [177, 183, 184].

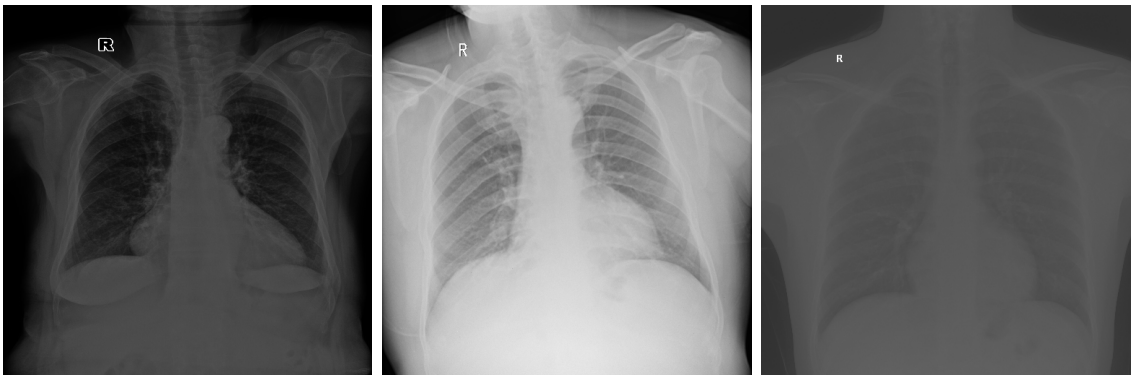
2.2.2.3 Domain Shift

As listed in Table 2.1, semi-supervised learning commonly assumes the labeled and unlabeled data come from either the same data source or two different data sources, *i.e.* there are only one domain or two domains. Mathematically, let $\mathcal{X}_L$ be the sample space for the labeled data and $\mathcal{X}_U$ be the sample space for the unlabeled data. $\mathcal{H}$-divergence is commonly used to define the domain divergence [5, 94, 13, 12]. Following the notation in the field of *domain adaptation* [59], I use $d_{\mathcal{H}}(\cdot, \cdot)$ to describe the domain divergence between two domains. When $\mathcal{X}_L$ and $\mathcal{X}_U$ represent two different data sources (*e.g.* two medical datasets collected from different hospitals), there exists *domain shift* or *dataset shift* [94, 13, 136] if $d_{\mathcal{H}}(\mathcal{X}_L, \mathcal{X}_U) > 0$. The domain shift in the medical imaging domain could arise from imaging protocols or modalities [38]. For example, chest X-rays taken by different scanners in different hospitals can exhibit clear visual difference, as shown in Figure 2.2. When domain shift is severe, the learned model overfits on the labeled training data and generalizes poorly on the unlabeled training data, causing inefficient training.

However, in practice, as there are much more unlabeled data than labeled data in semi-supervised learning, domain shift is often ignored when designing an algorithm. Unlike semi-supervised learning, the situation of PSL is more complex. Assume a partially labeled training set defined in Section 2.2.1 can be represented as $\mathcal{S}_{train} = \bigcup_{i=1}^{K} \mathcal{D}_i$. In practice, each $\mathcal{D}_i$ can be associated with a sample space $\mathcal{X}_i$, and $\forall i \neq j, d_{\mathcal{H}}(\mathcal{X}_i, \mathcal{X}_j) > 0$. This time, the learned model is expected to generalize well on all domains, which is practically difficult.

2.2.2.4 Partial Supervision

The fourth challenge comes from partial supervision. Compared with semi-supervised learning, this challenge is also unique to PSL. Assume the model of interest is a neural network, defined in Section 2.1.1, the loss function in Equation (2.2) is commonly



(a) (b) (c)

Figure 2.2: Illustration of domain shift in medical imaging. Three chest X-ray images collected from three different hospitals exhibit clear visual difference in brightness and contrast. Three chest X-ray images are selected from Wingspan Dataset [38].

minimized via *stochastic gradient descent* (SGD) [125] or variants of SGD such as mini-batch gradient descent. If $\mathbf{y}$ only contains partial supervision now, there is missing information with respect to the unannotated sub-tasks.

Using MLC as an example, a full label $\mathbf{y}$ is a C-element vector $[y^1, y^2, \cdots, y^C]^T$ with binary values, where $\forall 1 \leq i \leq C, y^i \in \{0, 1\}$. Assuming only the first class is annotated (*e.g.* $y^1 = 1$ or $y^1 = 0$), then all other classes ($y^{i>1}$) become null values. This means that in the stochastic gradient descent process, no gradients with respect to unannotated classes will be back-propagated. By replacing a full label with a partial label in Equation (2.5), only the value for $\mathcal{L}_{BCE}(g_1 \circ f_{shared}(x), y^1)$ can be computed given the input x. Then, in the backpropagation process, only f_{shared} and g_1 will receive the gradient update. In this case, MTL degenerates to STL and the advantages of inductive transfer between tasks can not be utilized. Moreover, f_{shared} might be biased towards the first class. This case can also be generalized where more than one class (task) is annotated. The same situations can also be observed in other MTL tasks with similar problem formulations to multi-label classification.

2.3 Partially Supervised Multi-Label Classification

2.3.1 Task Setup

In this thesis, the concept of PSL is illustrated with multi-label classification, a fundamental task in machine learning. The problem of interest is denoted as partially supervised multi-label classification (PSMLC). The mathematical formulation of PSMLC is similar to multi-label classification in Section 2.1.3. The only difference is the loss

function. Given an instance-label pair $(x, \mathbf{y})$, where $\mathbf{y}$ is a partial label with respect to x, the loss for PSMLC is

$$\mathcal{L}_{PSMLC}(x, \mathbf{y}) = \sum_{t \in \mathcal{T}_L(x)} \mathcal{L}_{BCE}(g_t \circ f_{shared}(x), y^t), \tag{2.6}$$

where $\mathcal{T}_L(x) \subseteq \mathcal{T}$ denotes the labeled set of tasks for x. When $\mathcal{T}_L(x) == \mathcal{T}$, *i.e.* all tasks are labeled for x, Equation (2.6) is equivalent to Equation (2.5).

It is worth mentioning that this thesis specifically focuses on the data scarcity (Section 2.2.2.1) scenarios as large-scale training data can alleviate the negative impact of partial supervision, *i.e.* each parameter can receive enough gradient updates when the training set is large enough.

2.3.2 Evaluation Metrics

The evaluation strategy of PSMLC is identical to binary classification, the sub-task of standard multi-label classification. The basic evaluation metric is the binary classification accuracy.

$$Acc = \frac{n_c}{n} \tag{2.7}$$

In Equation 2.7, n_c is the number of correctly classified examples and n is the number of total examples. A higher accuracy indicates a better classification performance.

The error rate is a metric conjugate to accuracy, where the sum of accuracy and error rate is 1. A lower error rate indicates a better classification performance.

$$Err = 1 - Acc = 1 - \frac{n_c}{n} \tag{2.8}$$

Both accuracy and error rate range from 0 to 1. However, accuracy and error rate are not robust metrics under class imbalance (Section 2.2.2.2). Thus, two more reliable metrics are considered in this thesis, which are F1 score and AUROC.

2.3.2.1 F1 Score

The first evaluation metric of interest is F1 score [147]. F1 score, also known as F-measure, or balanced F-score, is an evaluation metric for classification tasks. The score ranges from 0 to 1, where 0 is the worst and 1 is the best possible score. F1 score provides robust results for imbalanced data and evaluates both the precision

and recall ability of the model. Mathematically, F1 score is the harmonic mean of precision and recall.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.9}$$

Precision denotes the fraction of correctly predicted positive examples among all predicted positive examples.

$$precision = \frac{TP}{TP + FP} \tag{2.10}$$

In Equation (2.10), "TP" (true positive) denotes the number of positive examples that are correctly predicted as positive and "FP" (false positive) denotes the number of negative examples that are incorrectly predicted as positive. Similarly, recall denotes the fraction of correctly predicted positive examples among all ground truth positive examples. *i.e.*

$$recall = \frac{TP}{TP + FN} \tag{2.11}$$

In Equation (2.11), "TP" denotes the number of positive examples that are correctly predicted as positive and "FN" (false negative) denotes the number of positive examples that are incorrectly predicted as negative.

The implementation of F1 score leverages the function `sklearn.metrics.f1_score` from the `scikit-learn` package. [2]

2.3.2.2 AUROC

The second evaluation metric of interest is the *area under the receiver operating characteristic curve* (AUROC) [118]. The score it produces ranges from 0 to 1 where 0 is the worst score and 1 is the best score. Note, 0.5 means the model is as good as random. The metric is calculated as the area underneath the *receiver operating characteristic* (ROC) curve. The ROC curve is a graph which maps the relationship between the true positive rate of the model and the false positive rate. The true positive rate (TPR) is just the recall in Equation (2.11). The false positive rate (FPR) is calculated as the ratio between the number of negative examples wrongly categorized as positive examples and the total number of actual negative examples.

$$FPR = \frac{FP}{FP + TN} \tag{2.12}$$

In Equation (2.12), "TN" denotes the number of negative examples that are correctly predicted as negative. The ROC curve shows at various intervals the true positive

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html`

rate that the model is expected to achieve for a given trade-off with false positive rate. The area under this ROC curve, AUROC, therefore equates to the model's ability to predict classes correctly, as a large amount of area would show that the model can achieve a high true positive rate with a correspondingly low false positive rate.

The implementation of AUROC leverages the function `sklearn.metrics.roc_auc_score` from `scikit-learn` package.[3].

2.3.2.3 Choice of Evaluation Metric

Both F1 score and AUROC are common evaluation metrics in multi-label classification. There are two key differences between F1 score and AUROC. First, the inputs they take are different. First, F1 score requires predicted classes whilst AUROC only needs the predicted probabilities. Thus, before using F1 score, a threshold shall be defined as a probability boundary, *i.e.* $threshold \in (0, 1)$. In other words, F1 score is dependent on the choice of threshold. In this thesis, the threshold is default as 0.5. Second, two metrics handle imbalanced datasets differently. AUROC does not perform well on imbalanced datasets which often leads to misleading results, whilst F1 score is still able to measure performance objectively when the class balance is skewed.

In this thesis, I follow the common practice [177, 140] in the machine learning community when selecting the evaluation metric for the task of interest in each chapter. That is to say, the choice of evaluation metric is task-dependent. Specifically, I use F1 score when there are only few sub-tasks (classes) of interest and the intra-class distribution is imbalanced. In this scenario, I focus on the model performance of individual classes. I use AUROC when there are multiple sub-tasks (classes) of interest and the intra-class distribution is relatively balanced. In this scenario, I focus on the overall model performance.

2.4 Literature Review

2.4.1 Partially Supervised Learning in Medical Domain

PSL has become an emerging research direction for label-efficient learning. Recent studies focus on the medical applications where efforts have been made to utilize multiple partially labeled datasets for multi-organ/structure segmentation. Petit *et al.* [134] propose Semantic segmentation with MIssing Labels and ConvNEts

[3] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html`

(SMILE), a convolutional neural network to address the issue of learning with partial labels. SMILE tackles the partial labels issue by identifying and ignoring ambiguous labels during training, *i.e.* incorrect or noisy label information are not propagated in SGD. Petit *et al*. [134] also proposes SMILEr, which utilizes SMILE as initialization to generate pseudo labels in a curriculum learning [15] fashion. SMILE and SMILEr [134] are evaluated with liver, stomach, and pancreas segmentation tasks on CT scans. González *et al*. [63] adopts a similar learning strategy as [134] by proposing a cost-function that ignores unlabeled voxels in CT scans. The method proposed by [63] is evaluated in the context of pectoralis muscle and subcutaneous fat segmentation. Zhou *et al*. [199] also address the background ambiguity caused by partial supervision and proposes Prior-aware Neural Network (PaNN) which explicitly incorporates anatomical priors. PaNN makes an assumption that organ size distributions across datasets should be similar and utilizes a fully-labeled dataset to obtain prior statistics. [199] uses multi-organ segmentation on abdominal CT scans as evaluation. Fang *et al*. [54] propose a target adaptive loss (TAL) that allows training a segmentation network over multiple partially labeled datasets.The core idea of TAL is to treat the unknown labels as background. Shi *et al*. [150] propose two loss functions, namely marginal loss and exclusion loss, for multi-organ segmentation tasks. By utilizing fully labeled data, marginal loss merges all unlabeled pixels into the background pixels, same as TAL. Exclusion loss utilizes the prior information that organs are mutually exclusive. Inspired by conditional generative adversarial nets [120], Zhang *et al*. [191] propose a conditional version of nnU-Net [84] by leveraging prior conditional information for 3D multi-organ abdominal CT segmentation. Instead of solving multiple segmentation sub-tasks at the same time, [193] only segments each organ separately. Meanwhile, similar to [191], [193] encodes the information of the current segmentation task as a task-aware prior.

2.4.2 Limitations of Existing Studies

However, none of the aforementioned studies in Section 2.4.1 specifically take the first challenges in Section 2.2.2, namely data scarcity, class imbalance, and domain shift, into consideration. For example, all these methods assume that large-scale partially labeled data are available. Among these methods, [199, 150] additionally require fully labeled data, which are unrealistic to acquire under certain situations.

Existing studies aim to provide practical solutions to specific medical image applications by designing complex neural network architectures [54, 193] or training procedures [134, 199, 150, 191]. However, this does not directly help the community

to understand the fundamental challenges of PSL, especially under data scarcity. As the primary research question to answer in this thesis is to formulate PSL under challenging data scarcity setups, this thesis focuses on the learning paradigm, *i.e.* how to leverage partial labels in the learning process. Specifically, in Chapter 3, only small-scale partially labeled data are available; in Chapter 4, there are additional unlabeled data; in Chapter 5 and Chapter 6, partially labeled datasets are stored in a decentralized fashion. These problem setups are unexplored before. For each of these challenging setup, a novel solution is proposed for PSMLC, the task of interest defined in Section 2.3. In addition to the PSL studies discussed here, the relevant state-of-the-art are reviewed separately in each chapter.

2.5 Summary

This chapter first provides the necessary background knowledge for this thesis, including MTL, semi-supervised learning, and MLC. Then, I formalize the main concepts to understand PSL and summarize the main challenges for PSL. Then, I introduce PSMLC, the task of interest, which will be used to illustrate the problems of PSL under data scarcity throughout this thesis. Last but not least, I review the literature and discuss the limitations of existing studies.

Chapter 3

Partially Supervised Learning with Vicinal Risk Minimization

As discussed in Chapter 2, partial supervision and data scarcity are two major challenges of partially supervised learning. In classical machine learning, data augmentation is a practical tool to improve model generalization and robustness, especially under data scarcity [101, 28]. However, standard data augmentation [21] does not address the issue of partial supervision. For the first time, this chapter offers a simple yet efficient data augmentation solution to partially supervised learning under data scarcity by augmenting the input images and partial labels at the same time.

3.1 Introduction

In partially supervised learning (PSL), a naive solution to the challenge of *partial supervision*, discussed in Section 2.2.2.1, is to generate pseudo labels for the unlabeled sub-tasks under the formulation of multi-task learning. However, common label generation techniques for semi-supervised learning such as label propagation [201] and consistency regularization [157] require enough labeled data to generated high-quality pseudo labels. For example, existing PSL studies [134, 63, 50, 199, 54, 150, 182, 193] tend to assume that large-scale partially labeled or even fully labeled data are available during the training process. However, this is infeasible in many specific scenarios, especially in the medical domain, where the challenge of *data scarcity*, discussed in Section 2.2.2.1, has become a topic of active research.

Recall Section 2.2.1, given an instance with multiple sub-tasks of interest, an instance has *full* labels if all the sub-tasks have annotations and an instance has *partial* labels with respect to all the sub-tasks if only a subset of sub-tasks have annotations. Take multi-label classification as an example. A medical image can be

associated with multiple diseases as the same time and it can be partially labeled for a few diseases and unlabeled for the rest of diseases.

While learning to generate pseudo labels has a demanding requirement on the number of available ground truth labels, a different strategy is proposed in this chapter. I leverage vicinal risk minimization (VRM) [19] to address the challenges of partial supervision and data scarcity in PSL at the same time. Instead of learning a *discriminative* model to generate pseudo labels for unlabeled tasks, *vicinal* instance-label pairs are generated by a *bounded* vicinal distribution. Here, "vicinal" means that the generated instance-label pairs are in the vicinity of ground truth instance-label pairs. A detailed explanation of VRM is presented in Section 3.3.1. With vicinal instance-label pairs, a partially supervised problem is transformed into a supervised one and standard supervised learning techniques can be applied [39].

VRM is a family of data augmentation methods that generate vicinal instance-label pairs given ground truth instance-label pairs. Under the normal setup of supervised learning, VRM has been proved as an efficient data augmentation technique to improve the model generalization [192, 188, 135, 96, 95, 165]. Let us take MixUp [192] as an example. In supervised learning, MixUp generates a vicinal instance-label pair by linearly adding two ground truth instance-label pairs, where the ground truth instance-label pairs have all sub-tasks of interest labeled (see Section 3.3.2 for detailed description of MixUp). However, the application of VRM in PSL remains an open question. The challenges of applying VRM to partially supervised multi-label classification (PSMLC) are discussed in Section 3.4 in detail.

In this chapter, a simple yet robust MixUp-based method is proposed to PSMLC under data scarcity. Similar to MixUp, the vicinal image is generated by linearly adding two images. While MixUp requires that two images have two full labels respectively, the proposed method can relax this constraint: given a sub-task of interest, only one of two instance is required to have a label for this sub-task. The constructed vicinal distribution is regularized by a prior distribution inspired by the principle of maximum entropy (PME) [87] to bound the vicinal label space. Specifically, a pseudo label is assigned to the unlabeled instance (with respect to the task of interest) based on a prior distribution. Then, the ground truth label and the pseudo label are linearly added to form the vicinal label. A theoretical guarantee is further provided for the quality of the generated vicinal label. The experiments show that, given a neural network, the proposed method can efficiently improve the model generalization, with access to only partial labels, and is robust under data scarcity and class imbalance.

The main contributions of this chapter can be summarized as follows:

1. This is the first study of partially supervised multi-label classification (PSMLC) under data scarcity in the medical domain.

2. A simple yet robust technique based on vicinal risk minimization is proposed to improve the model performance with only scarce partially labeled medical images.

3. A theoretical analysis is provided for the proposed method.

4. The experimental results show initial empirical insights into future research directions on PSL under data scarcity.

3.2 Related Work

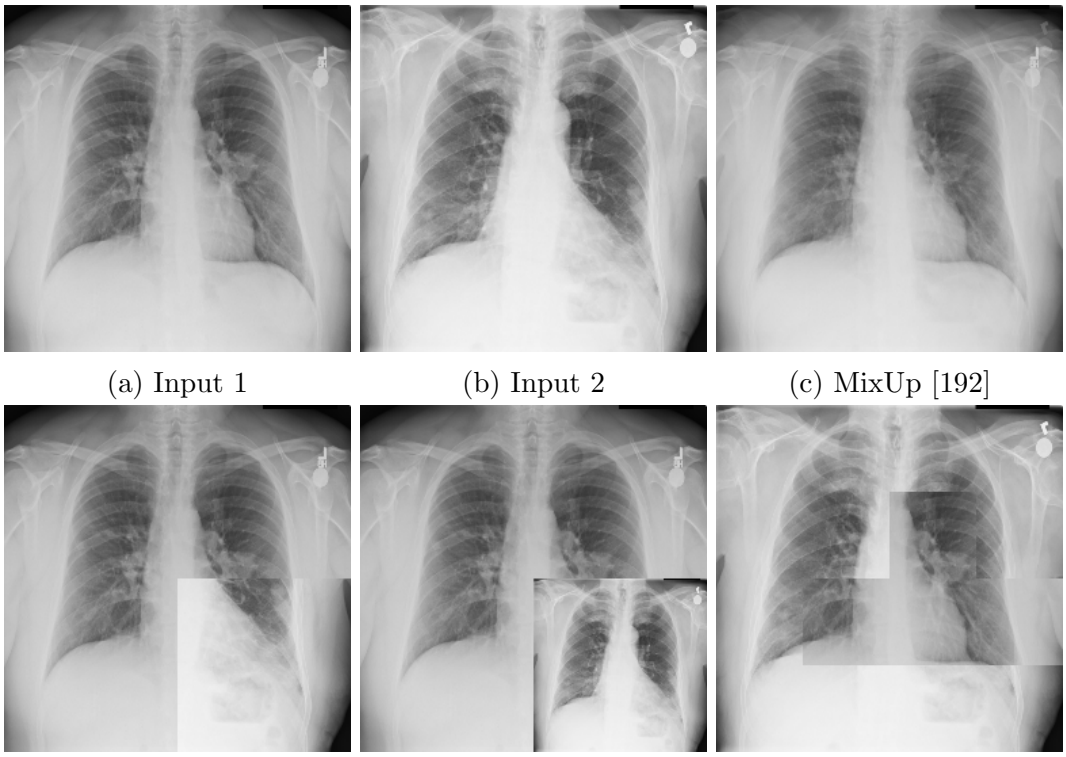
3.2.1 Vicinal Risk Minimization

The core idea of vicinal risk minimization (VRM) is to construct a vicinal distribution based on the training set. The state-of-the-art VRM methods are illustrated in Figure 3.1 (two inputs are visually similar) and Figure 3.2 (two inputs are visually different). Zhang *et al.* [192] first proposes MixUp (Figure 3.1c and Figure 3.2c), a data augmentation method in the input space by mixing two random images and the corresponding labels by convex interpolation. This idea has been further extended by many variants. For example, CutMix [188] (Figure 3.1d and Figure 3.2d) replaces a random patch of the first image with a patch from the second image cropped at the same location. ResizeMix [135] (Figure 3.1e and Figure 3.2e) takes one step further by replacing a random patch of the first image with the resized second image. PuzzleMix [96] (Figure 3.1f and Figure 3.2f) could be viewed as a generalization of CutMix, where the locations of the random patches are determined by saliency.

These VRM-based data augmentation techniques [192, 188, 135, 96, 95, 165] have been designed to successfully improve the generalization ability of a standard multi-class classifier trained on general object-centric images with full labels. See Figure 3.3 for visual differences between general object-centric images and multi-object medical images. However, to apply VRM to PSMLC, the problem of interest, there are still several challenges to be addressed, which will be discussed in Section 3.4.

3.2.2 Partially Supervised Multi-Label Classification

The problem formulation of PSMLC has been introduced in Section 2.3. So far, only limited efforts [50] have been made to fully understand the problem, especially under



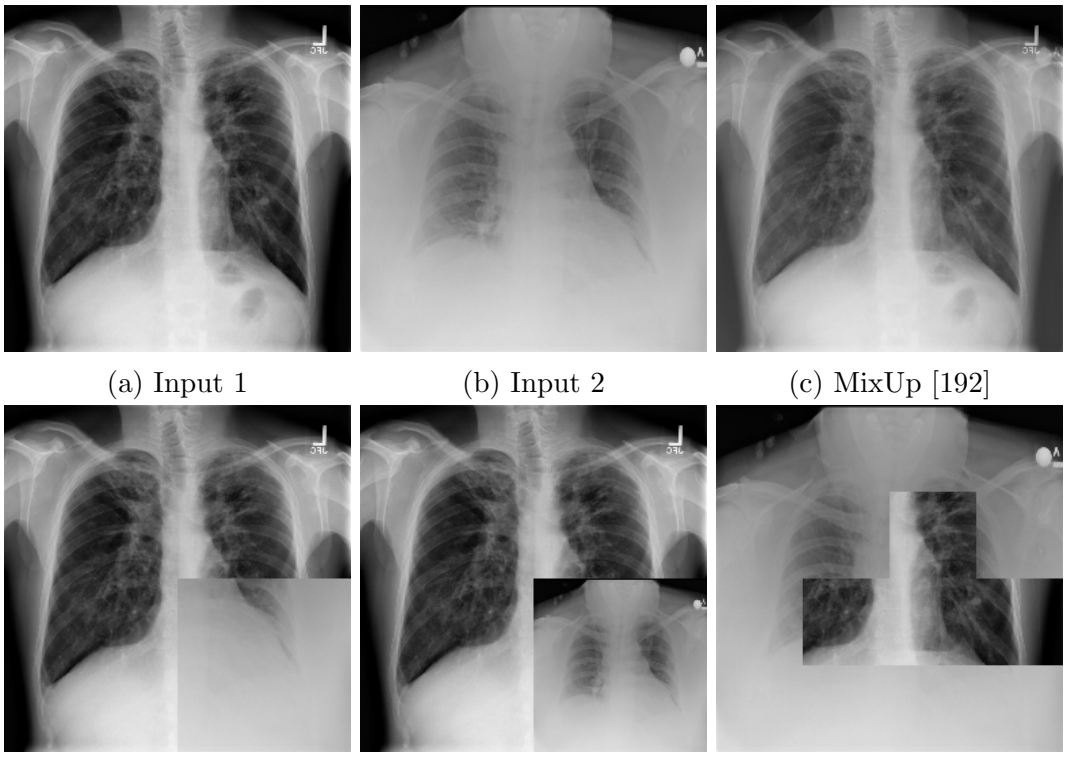
(a) Input 1 (b) Input 2 (c) MixUp [192]

(d) CutMix [188] (e) ResizeMix [135] (f) PuzzleMix [96]

Figure 3.1: Illustration of vicinal images generated by state-of-the-art VRM methods on two visually similar input chest X-ray images. (a) and (b) are input images, while (c) - (d) are vicinal images. The mixing ratio $\lambda = 0.75$ is used for all four methods displayed and the value is chosen for illustration purpose. See Section 3.3.2 for the definition of λ.

data scarcity. Durand *et al.* [50] tackle the problem from the perspective of label propagation [201] with access to large-scale benchmark datasets. With abundant partially labeled data, the problem can be factually linked with standard supervised learning or semi-supervised learning. Thus, the problem formulation of data scarcity in this study differentiates my contributions from those proposed by Durand *et al.* in [50].

As mentioned in Section 3.1, it is difficult to learn high-quality pseudo labels for unlabeled tasks for existing PSL methods [199, 39] under data scarcity. Thus, the goal of this chapter is to generate robust pseudo labels with only limited partial labels. To achieve this goal, the problem of interest is reformulated as a VRM problem, where pseudo labels are generated from an uncertainty-bounded vicinal distribution.



(a) Input 1 (b) Input 2 (c) MixUp [192]

(d) CutMix [188] (e) ResizeMix [135] (f) PuzzleMix [96]

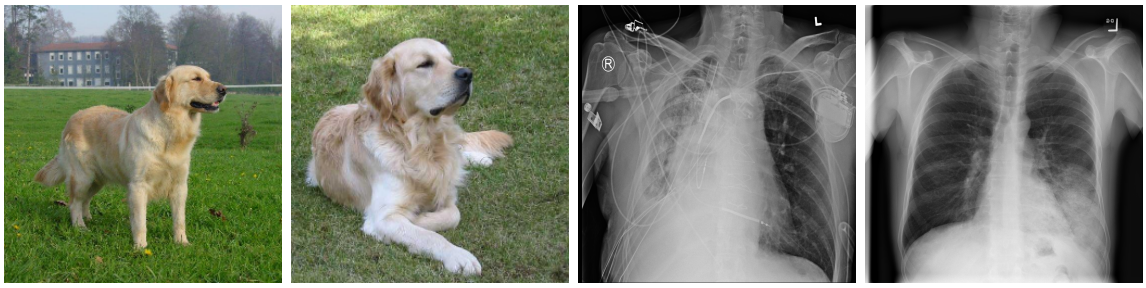
Figure 3.2: Illustration of vicinal images generated by state-of-the-art VRM methods on two visually different input chest X-ray images. (a) and (b) are input images, while (c) - (d) are vicinal images. The mixing ratio $\lambda = 0.75$ is used for all four methods displayed and the value is chosen for illustration purpose. See Section 3.3.2 for the definition of λ.

3.3 Preliminaries

3.3.1 Vicinal Risk Minimization

The concept of vicinal risk minimization (VRM) is an extension of empirical risk minimization (ERM). In ERM, the risk (loss function) with respect to an empirical distribution of ground truth instance-label pair (x, y) is minimized. For example, the training set $\mathcal{S}_{train}$ and the test set $\mathcal{S}_{test}$ are sampled from this empirical distribution. In VRM, a vicinity distribution $\mathcal{V}$ is defined as a distribution for the virtual instance-label pair (also known as vicinal instance-label pair) $(\tilde{x}, \tilde{y})$. The term "vicinal" comes from the fact that $(\tilde{x}, \tilde{y})$ should lie in the vicinity of ground truth instance-label pair $(x, y) \in \mathcal{S}_{train}$. The *vicinal risk* (*c.f.* empirical risk) is defined as

$$\mathcal{R}_{\mathcal{V}}(h) = \frac{1}{|\mathcal{V}|} \sum_{(\tilde{x}_i, \tilde{y}_i) \in \mathcal{V}} L(h(\tilde{x}_i), \tilde{y}_i), \tag{3.1}$$



(a) (b) (c) (d)

Figure 3.3: (a) and (b) are two golden retrievers from ImageNet [34]. (c) and (d) are two chest X-ray images from ChestX-ray14 [177]. In contrast to object-centric images, chest X-ray images have multiple objects of semantic interests (*e.g.* organs) within the same image.

where $\mathcal{L}(\cdot, \cdot)$ is the loss function and $h \in \mathcal{H}$ is the hypothesis. In Equation (3.1), L and h are assumed to be universal, *i.e.* they can be any loss function and model in a standard supervised setting.

3.3.2 MixUp

MixUp [192] is a data augmentation method which takes two instance-label pairs as input and generates a virtual instance-label pair. Let (x_i, y_i) and (x_j, y_j) be two instance-label pairs randomly sampled from the training set $\mathcal{S}_{train} = \{(x_k, y_k)\}_{k=1}^{n}$. Let $x \in \mathbb{R}^{H \times W \times 3}$ be an image and $y \in \{0, 1\}^C$ be the corresponding one-hot encoded vector with respect to x, where there are C classes of interest. The vicinal image-label pair $(\tilde{x}, \tilde{y})$ is defined as

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1 - \lambda) x_j \\ \tilde{y} &= \lambda y_i + (1 - \lambda) y_j \end{aligned} \tag{3.2}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$ is the mixing ratio and $\alpha \in (0, \infty)$ is a hyperparameter for the beta distribution. Note, Equation (3.2) can be applied directly to both multi-class classification and multi-label classification problem. From the perspective of multi-task learning [17], a C-class multi-label classification task can also be decomposed into C binary classification tasks.

3.3.3 Weighted Loss

To combat class imbalance, the weighted binary cross-entropy (BCE) loss is commonly adopted in multi-label classification tasks. Given an instance-label pair (x, y), y^i

denotes the i^{th} entry of the label vector y. The weighted binary cross-entropy loss for the i^{th} class is

$$\mathcal{L}_{BCE}(x, y^i, w_+^i, w_-^i) = -w_+^i y^i \log p(y^i = 1|x) - w_-^i (1 - y^i) \log p(y^i = 0|x), \quad (3.3)$$

where $w_+^i = \frac{n_-^i}{n_+^i + n_-^i}$ and $w_-^i = \frac{n_+^i}{n_+^i + n_-^i}$ with n_+^i and n_-^i the number of positive and negative cases for the i^{th} class, respectively. In contrast to Equation (2.4), Equation (3.3) has additional weights w_+^i and w_-^i.

Analogous to Equation (2.5), the weighted loss for multi-label classification (MLC) is then

$$\mathcal{L}_{MLC}(x, y) = \sum_{i=1}^{C} \mathcal{L}_{BCE}(x, y^i, w_+^i, w_-^i). \quad (3.4)$$

3.4 Challenges of Vicinal Risk Minimization for Partially Supervised Multi-Label Classification

Before presenting the proposed method, let us first review the challenges of applying VRM on PSMLC.

3.4.1 Challenge of Generating Vicinal Images

The first challenge is to generate vicinal images. Though many of these methods [188, 135, 96] have reported state-of-the-art performance in multi-class classification tasks on general images, they tend to generate less meaningful vicinal images than MixUp in the medical domain. This phenomenon is due to the fact that these methods are designed for object-centric images where the mixing process can potentially keep the semantic information of interest. There are three observations from Figure 3.1 and Figure 3.2. Firstly, compared with MixUp [192], MixUp variants such as CutMix [188], ResizeMix [135], and PuzzleMix [96] generate less perceptually comfortable vicinal images (*i.e.* they do not look like real medical images). Secondly, CutMix, ResizeMix, and PuzzleMix do not utilize the *human structure similarity* [38, 47, 40, 39]. Last but not least, in a multi-label classification problem scenario, the vicinal images generated by CutMix, ResizeMix, and PuzzleMix might discard potential region of interests. For example, if the diseased region for the first input lies in the bottom right part of the chest X-ray image, this might not be preserved in the vicinal images. This issue is demonstrated in Figure 3.1 by contrasting 3.1c against Figure 3.1d, Figure 3.1e, and Figure 3.1f and in Figure 3.2 by contrasting 3.2c against Figure 3.2d, Figure 3.2e, and Figure 3.2f. Overall, MixUp shows robust

visual performance over the other variants. Thus, MixUp is adopted as the baseline to generate vicinal images in this chapter.

3.4.2 Challenge of Generating Vicinal Labels

The second challenge is to generate vicinal labels. The existing VRM methods are only designed for fully labeled data. Due to partial supervision, each instance may have unlabeled tasks, *i.e.* only partial labels are available. That is to say, while two images can be randomly sampled to generate a vicinal image with MixUp, the corresponding vicinal label is ill-defined with missing labels. This is also deemed as the major bottleneck of applying VRM to PSMLC.

3.5 Method

For simplicity, the proposed method is illustrated with the base case of two classes of interest ($C = 2$), while more general cases (*i.e.* $C > 2$) can be easily derived based on the base case. Again, (x_i, y_i) and (x_j, y_j) are two instance-label pairs, where y_i and y_j are partial labels. There are two cases regarding the partial labels.

3.5.1 Locally Full Supervision

In the first case, y_i and y_j have partial labels for the same class. Concretely, a possible situation is $y_i = [?, y_i^2]$ and $y_j = [?, y_j^2]$, where ? denotes the null value of the missing label for the 1^{st} class and $y_i^2 \in \{0, 1\}, y_j^2 \in \{0, 1\}$ are the label regarding the 2^{nd} class. Obviously, Equation (3.2) still holds if only the 2^{nd} class is considered alone. Under the interpretation of multi-task learning, the 2^{nd} class now has *locally* full supervision. However, to leverage MixUp in this case, there will be additional computational costs in batch-wise sampling, which could be a non-trivial overhead in practice. A trivial solution is to decompose a PSMLC problem into multiple binary classification problems. However, when the number of classes C is large, this strategy will be inefficient as it does not utilize multi-task learning.

3.5.2 Globally Partial Supervision

The second case is more challenging than the first case. Here, y_i and y_j have partial labels for different classes, respectively. A concrete example might be $y_i = [?, y_i^2]$ and $y_j = [y_j^1, ?]$ or $y_i = [y_i^1, ?]$ and $y_j = [?, y_j^2]$. When the partially labeled dataset consists of partially labeled data collected from various sources, the majority of randomly

sampled pairs will have partial labels for different classes. This is also the major bottleneck of PSMLC.

The proposed solution is to transform the partially supervised problem into a fully supervised one. In VRM, the first priority is to define the vicinity distribution [19]. With missing ground truth label information, this is infeasible for PSMLC as there is too much uncertainty. Intuitively, the solution is to use inductive bias (prior belief) to bound the uncertainty when constructing the vicinity distribution. Here is an example, where the vicinity distribution is regularized with a simple probability trick.

The principle of maximum entropy (PME) [87] was first proposed in 1950s. With limited prior knowledge over the unknown true distribution, PME defines the target distribution with the largest entropy. Concretely, for the k^{th} class, assume only x_i has the partial label (*e.g.* $y_i^k = 1$) and x_j has a missing label ($y_j^k \leftarrow ?$). Considering y_j^k as an independent system with two possible states, a discrete uniform distribution leads to the largest entropy. So, without any prior knowledge about x_j, it is possible to define $\hat{y}_j^k = 0.5$. The vicinity distribution is defined as

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda) x_j \\ \tilde{y}^k &= \lambda y_i^k + (1-\lambda)\hat{y}_j^k, \end{aligned} \tag{3.5}$$

where $\lambda \sim \text{Uniform}(\alpha_k, 1)$ follows a uniform distribution and $0.5 \leq \alpha_k < 1$ is a hyperparameter. Note, α_k, instead of α, is used in the $\text{Uniform}(\alpha_k, 1)$ as the choice of α_k can be dependent on the k^{th} class. This is different from MixUp-based methods [192, 188, 135, 96, 95, 165] that are designed for multi-class classification tasks. More details of this design are discussed in Section 3.7.2.4. Though Equation (3.5) has a similar format to Equation (3.2), a few differences should be highlighted: (a) x_i should have locally full supervision for the k^{th} class (*e.g.* y_i^k=1 or y_i^k=0); (b) λ is required to be greater than 0.5 for the known example. Clauses (a) and (b) ensure that the generated vicinal label $\tilde{y}^k$ has a logically reasonable label distribution, which is analyzed in Section 3.6.1. An illustrative comparison between MixUp (when full labels are available) and the proposed method (when only partial labels are available) is presented in Figure 3.4.

3.5.3 Training Strategy

For fully labeled images, the linear combination in Equation (3.2) can be performed efficiently with batch-wise processing, as shown in [192]. However, it requires additional computational cost to sample two image-label pairs to apply Equation (3.2) in PSMLC. First, the task has to be decomposed into C binary classification sub-tasks,

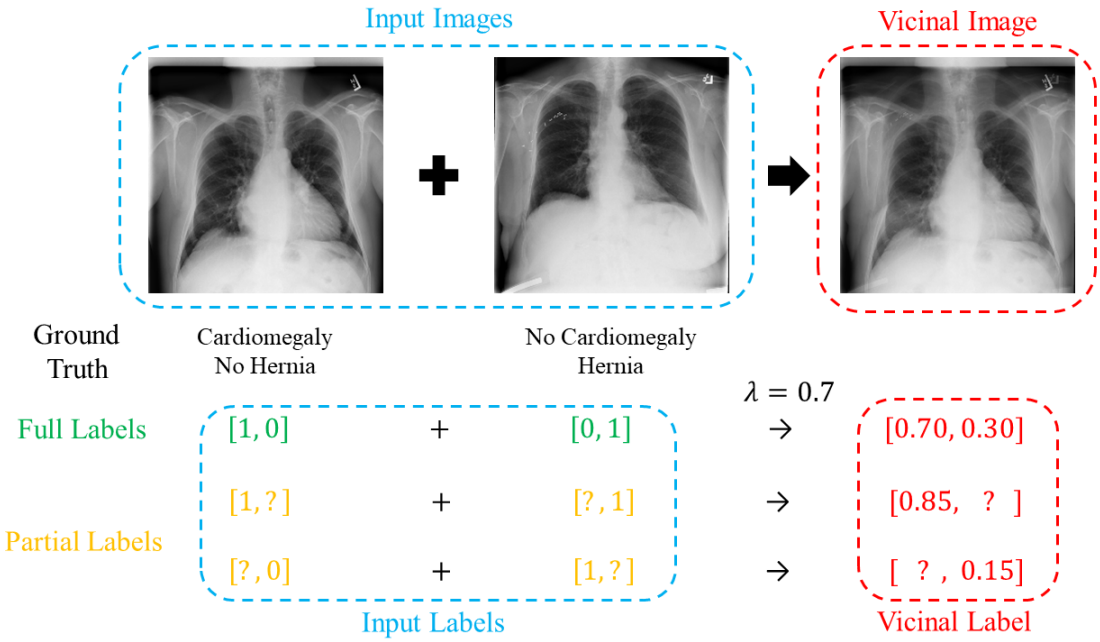


Figure 3.4: Generating a vicinal image and a vicinal label by the proposed method. The input images are two chest X-ray images. There are two classes of interest, which are cardiomegaly and hernia. The binary label "1" denotes that the image contains the class of interest and "0" denotes that the image does not contain the class. For example, for the first image, $[1, 0]$ means that the image only contains the symptoms of cardiomegaly and does not have signs of hernia. λ is set as 0.7. When full labels are available for both images, Equation (3.2) can be directly applied to generate the vicinal labels for all classes of interest. Two possible situations are depicted when both images are partially labeled. "?" denotes the missing binary label. Given a specific class, as long as the first image has a ground truth label for it, Equation (3.5) can still generate a vicinal label for this class.

which gives up the formulation of multi-task learning and increases the number of forward and backward passes in the optimization process.[1] Second, when sampling a pair, two images are required to be partially labeled for the same class of interest. Third, when C is large, two image-label pairs can have both locally full supervision and globally partial supervision. To fully utilize the advantages of random sampling in stochastic gradient descent, I need to efficiently implement Equation (3.2) and Equation (3.5) at the same time in the batch processing.

To solve the above issues, a computation-efficient implementation for batch-wise training with only 8 lines of PyTorch [132] code is presented, shown in Listing 3.1.[2]

[1] In backpropagation, the forward pass computes the output of the network given the input data and the backward pass computes the output error with respect to the expected output and then goes backward into the neural network and updates the model weights using gradient descent.

[2] For simplicity, I use the same value of `alpha` for different classes to illustrate the main concept

It is worth mentioning that the proposed method works if $C > 1$. Here, C is also the length of the label vector. Thus, given two input label vector, the vicinal label vector is just a linear combination two label vectors as long as two label vector have the same length C. For a better illustration of the mechanism behind the proposed implementation, a concrete example with $C = 4$, with a step-by-step demonstration of Listing 3.1 is provided in Table 3.1. Let $y_1 = [1, ?, 0, ?]$ and $y_2 = [1, 1, ?, ?]$ be two label vectors with missing labels ("?"). Given an arbitrary $\lambda \sim \text{Uniform}(0.5, 1)$ (say $\lambda = 0.75$), I efficiently acquire the logits and (vicinal) labels to compute the binary cross entropy loss in Equation (3.3).

```
# x1, x2: batch of images, [B, 3, H, W]
# y1, y2: batch partial labels where the missing
#         entries are filled with 0.5, [B, K]
# alpha: hyperparameter
# model: neural network
# criterion: loss function with reduction='none'
lam = numpy.random.uniform(alpha, 1)
mask = y1 != 0.5
x = lam * x1 + (1. - lam) * x2
y = lam * y1 + (1. - lam) * y2
loss = (criterion(model(x), y) * mask).mean()
optimizer.zero_grad()
loss.backward()
optimizer.step()
```

Listing 3.1: PyTorch code for batch-wise training.

Table 3.1: A step-by-step demonstration of Listing 3.1. After getting $\tilde{y}$ with Equation (3.2) and/or Equation (3.5) with $\lambda = 0.75$, the binary cross entropy losses for $C = 4$ classes are computed following Equation (3.3). However, only the binary cross entropy losses of the first and the third classes will be back-propagated as the binary cross entropy losses of the second and the fourth classes are zeroed out. As a comparison, MixUp can only be applied to the first class, when both input labels are available.

Step	y_1	y_2	mask
–	[1, ?, 0, ?]	[1, 1, ?, ?]	–
Fill ? with 0.5	[1, 0.5, 0, 0.5]	[1, 1, 0.5, 0.5]	–
Get mask	[1, 0.5, 0, 0.5]	[1, 1, 0.5, 0.5]	[1, 0, 1, 0]
Get $\tilde{y}$	[1, 0.625, 0.25, 0.5]		[1, 0, 1, 0]

of the proposed method.

	$y_i^k = 1$	$y_i^k = 0$
$y_j^k = 1$	1	$1 - \lambda$
$y_j^k = 0$	λ	0

Table 3.2: $\tilde{y}^k$ generated by Equation (3.2) when both y_i^k and y_j^k are known.

	$y_i^k = 1$	$y_i^k = 0$
$y_j^k = 1$	$0.5(1+\lambda)$	$0.5(1-\lambda)$
$y_j^k = 0$	$0.5(1+\lambda)$	$0.5(1-\lambda)$

Table 3.3: $\tilde{y}^k$ generated by Equation (3.5) when only y_i^k is known.

	$y_i^k = 1$	$y_i^k = 0$
$y_j^k = 1$	$0.5(1-\lambda)$	$0.5(1-\lambda)$
$y_j^k = 0$	$0.5(1-\lambda)$	$0.5(1-\lambda)$

Table 3.4: $\tilde{y}^k$ generated by Equation (3.5) when only y_i^k is known.

3.6 Theoretical Analysis

3.6.1 Theoretical Guarantee

It is important to have a theoretical guarantee for the quality of the generated vicinal labels. To achieve this, let us take a close look at the possible values of $\tilde{y}^k$. As a reference, the vicinal labels generated by Equation (3.2) when both y_i^k and y_j^k are known are summarized in Table 3.2. As shown in Table 3.2, y_i^k and y_j^k both have binary values ("0" or "1"). In total, there are four possible combinations of y_i^k and y_j^k. These values can be considered as "ground truth" values for the vicinal label $\tilde{y}^k$ under four possible states of (y_i^k, y_j^k).

Assume only y_i^k is known. It can be inferred from Equation (3.5) that if y_i^k=1, $0.75 \leq \tilde{y}^k = 0.5(1+\lambda) < 1$; if $y_i^k = 0$, $0 < \tilde{y}^k = 0.5(1-\lambda) \leq 0.25$. Similar to Table 3.2, the corresponding vicinal labels generated by Equation (3.5) are summarized in Table 3.3. For a quick sanity check, rounding the values in Table 3.2 and Table 3.3 to nearest integers, the two tables should be assigned with the same "virtual" binary labels (*i.e.* "0" or "1"). It is worth mentioning that the absolute differences between the corresponding values in Table 3.2 and Table 3.3 (summarized in Table 3.4) are the same ($0 < 0.5(1-\lambda) \leq 0.25$). As the values are bounded, the vicinity distribution is also bounded. In addition, because Equation (3.2) and Equation (3.5) lead to the same virtual binary labels, this difference can be interpreted as an instance of *label smoothing* [160].

3.6.2 Enlarged Sample Space

As MixUp is originally designed as a data augmentation method, a direct theoretical advantage of the proposed method is to have an enlarged sample space, which can mitigate the overfitting problem. To quantify this advantage, let us consider a simplified case. Assume there are C classes of interest and there are C partially labeled datasets, $\{\mathcal{D}_i\}_{i=1}^{C}$, where each dataset is labeled for one of C classes, respectively. Let $n_i = |\mathcal{D}_i|$ be the number of instances in $\mathcal{D}_i$. For *vanilla* supervised learning, which is the standard supervised learning without any data augmentation, there will be $\sum_i^C n_i$ different training instance-label pairs. Without considering the randomness effect caused by the uniform distribution, MixUp can generate $\sum_i^C n_i(n_i - 1)$ different training instance-label pairs, *i.e.* only considering locally full supervision with Equation (3.2). However, the proposed method can generate $\sum_i^C (n_i \sum_{j \neq i} n_j)$ different training instance-label pairs, which is much larger than the sample spaces of supervised learning and standard MixUp.

3.6.3 Computational Cost

The training process of the proposed method is almost identical to the training process for a fully supervised task, *i.e.* there is no additional optimization cost such as iterative training [199]. Similarly, the proposed method utilizes the same memory footprint in terms of network weights. As a comparison, a semi-supervised method such as label propagation, knowledge transfer, and knowledge distillation will require the training of multiple networks to generate pseudo-labels. For the proposed method, the major overheads arising from the data generation process are the random sampling, as mentioned in Section 3.5.3, and the element-wise operations on low-dimensional arrays in Equation (3.5), which are negligible compared to the backpropagation cost. Equation (3.5) can be easily implemented by any scientific computing frameworks supporting broadcasting, such as NumPy [71], PyTorch [132], or TensorFlow [1].

3.6.4 Relationship with Noisy Labels

Learning from noisy labels [127] has been a seminal strategy in utilizing unlabeled data over the past decade [201]. Here, the noisy labels are the pseudo labels that are automatically generated by the algorithms for the unlabeled data. This strategy has shown state-of-the-art performance in semi-supervised learning benchmark

Table 3.5: Comparison between learning-based pseudo labeling methods and the proposed method.

	Existing Methods	Proposed Method
generative label generation	✗	✓
augmentation on instances	✗	✓
feasibility under data scarcity	✗	✓

tasks [157]. The problem of interest shares a similar problem formulation with semi-supervised learning, where unlabeled data exist. However, the proposed method differs from semi-supervised learning in three aspects. First, instead of learning pseudo labels, the pseudo labels are generated by a bounded vicinity distribution. That means, the proposed method is computationally efficient. Second, essentially, the proposed method is a data augmentation technique, which simultaneously augments the base image (x_i in Equation (3.5)) and adds noise to the base label (y_i^k in Equation (3.5)). In fact, adding noise to the ground truth labels has been shown as an effective method to improve model generalization [167]. Last but not least, the generated pseudo labels ($\tilde{y}^k$) have a theoretical guarantee (Section 3.6.1). As shown in [39], semi-supervised learning-based noisy labels might not be reliable if there are not enough labeled data to learn the model. For the problem of interest, a semi-supervised learning approach is not feasible under a data scarcity situation. The differences between the proposed VRM-based method and existing learning-based pseudo-labeling methods are highlighted in Table 3.5.

3.7 Experiments

The purpose of the experimental design is twofold. First, there is no known empirical study of PSL under data scarcity. The impact of limited partial labels on deep learning models is investigated. Second, the proposed method is evaluated in a simulated environment to demonstrate its effectiveness under data scarcity.

3.7.1 Experimental Setup

3.7.1.1 Data

The public multi-label dataset of thoracic conditions ChestX-ray14 [177] is used in this study and its default batch splits are adopted to ensure the reproducibility. The dataset can be downloaded from `https://nihcc.app.box.com/v/ChestXray-NIHCC`. ChestX-ray14 is collected by National Institute of Health, United States. The dataset

contains chest X-ray data of 14 thoracic diseases, which are *atelectasis*, *cardiomegaly*, *consolidation*, *edema*, *effusion*, *emphysema*, *hernia*, *fibrosis*, *infiltration*, *nodule*, *pleural thickeningpneumonia*, and *pneumothorax*. The second 1000 chest X-ray images of the first batch are selected as the testing set. Without loss of generality, C is set as 4, aligned with the example shown in Table 3.1. In multi-label classification tasks, the classes are mutually exclusive. *i.e.* classes are independent. Thus, the empirical results generalize under different values C. Four most common diseases among 14 identified conditions are selected to form four partially labeled datasets, which are *infiltration*, *effusion*, *atelectasis*, and *nodule*.

3.7.1.2 Models

For simplicity, the proposed PME-based MixUp variant is denoted as MixUp-PME. This is a unification of locally full supervision and globally partial supervision, where Equation (3.2) and Equation (3.5) are efficiently integrated (as shown in Section 3.5.3).

Four baselines are considered in the experiments. The first one is a standard multi-label classification (MLC) model where the missing labels are ignored in the backpropagation [63]. This model is denoted as *vanilla*. For the second baseline, MixUp is applied in a locally full supervision fashion to train an MLC model, as described in Section 3.5.1. I use the default setup of MixUp [192], where α is set as 1. This is also the common practice in the literature. The third baseline is FixMatch [157], a state-of-the-art semi-supervised learning method that can be adapted to PSL. FixMatch incorporates pseudo labelling and knowledge distillation. The official implementation[3] is adapted to PSL in this study. To understand the impact of partial supervision under data scarcity, an *Oracle* is presented as the fourth baseline, which is the same MLC model trained with full labels. As expected, this is the best performance that the model of interest can achieve under standard supervised learning, which is also considered as the upper bound performance for centralized fully labeled data.

3.7.1.3 Implementation

All five models share the same network backbone. Following the setup of [140], DenseNet121 [81] is chosen as the network backbone. The weighted loss in Equation (3.4) is minimized by a standard Adam optimizer [97] with fixed learning rate 10^{-3} and batch size 64. In the inference phase, the default threshold for the predicted probability score is 0.5.

[3] `https://github.com/google-research/fixmatch`

In this chapter, F1 score (see Section 2.3.2.1 for details of F1 score) is used as the evaluation metric. The goal is to understand the impact of the proposed method on different classes and the intra-class imbalance and inter-class imbalance both exist (see Section 2.3.2.3 for the reasons). All models are trained for 30 epochs and the best mean F1 score for each model is reported. Though both mean and standard deviation are reported, mean is used as the primary metric when comparing the performance of the baselines in the problem of interest [177, 140, 192].

All experiments are conducted in PyTorch [132] on an NVIDIA Tesla V100. Note, the traditional data augmentation that manipulates the image space does not directly solve the partial supervision problem [39]. Thus, for a fair comparison, no traditional data augmentation is applied, except for FixMatch. All chest X-ray images are resized to a fixed size of 224×224. As a pre-processing step, instance normalization [30] is performed on each chest X-ray image:

$$\hat{x}^{ij} = \frac{x^{ij} - \mu(x)}{\sigma(x)} \tag{3.6}$$

where x is an image, $\hat{x}$ is the normalized image, (i, j) is the position of the pixel, and μ and σ are the mean and standard deviation of the pixels of x.

3.7.2 Empirical Analysis

3.7.2.1 Comparison with Baselines

In this experiment, the proposed method is illustrated with a simple case where each image is only labeled for one class. This situation can be interpreted as collecting C single-class datasets to build a multi-label classifier, which is the fundamental case. This fundamental case can also provide a controllable environment to study the impact of task-wise characteristics (*e.g.* the difficulty of each task). A complex situation that each image is partially labeled for multiple classes can be viewed as a trivial extension from the perspective of multi-task learning (MTL).

An inter-class balanced training set is created. For each of four classes, 100 chest X-ray images are sampled from the first 1000 chest X-ray images of the first batch of ChestX-ray14 to form a partially labeled sub-dataset. In each sub-dataset, there are 20 positive cases and 80 negative cases, *i.e.* the intra-class positive-negative ratio r is $\frac{1}{4} = \frac{20}{80}$. This leads to a training set of 400 images. In this experiment, the same value of $\alpha = 0.5$ is used for different classes. Each experiment is run three times with different random seeds. The mean and standard deviation of the F1 scores for model are presented in Table 3.6.

Table 3.6: Comparison with baselines. In the training set, each chest X-ray image is only partially labeled for one class. The proposed MixUp-PME achieves the best overall performance. The reported numbers are F1 scores. Blue denotes the highest number in each column (other than the *Oracle*).

Model	*Infiltration*	*Effusion*	*Atelectasis*	*Nodule*	Average
vanilla	0.2828 ± 0.0048	0.1042 ± 0.0547	0.1759 ± 0.0300	0.1482 ± 0.0635	0.1778 ± 0.0055
MixUp	0.2999 ± 0.0113	0.1362 ± 0.0135	0.1758 ± 0.0151	0.0933 ± 0.0115	0.1763 ± 0.0025
FixMatch	0.2358 ± 0.0728	0.1178 ± 0.0381	0.1624 ± 0.0152	0.1179 ± 0.0543	0.1583 ± 0.0246
MixUp-PME	0.3024 ± 0.0159	0.1433 ± 0.0025	0.1624 ± 0.0177	0.1667 ± 0.0827	0.1937 ± 0.0147
Oracle	0.3020 ± 0.0165	0.1305 ± 0.0202	0.1706 ± 0.0126	0.1955 ± 0.0588	0.1988 ± 0.0041

Though MixUp can achieve robust performance under standard supervised learning, it fails to improve the *vanilla* under PSL. Compared with *vanilla* and MixUp, MixUp-PME can significantly improve the overall performance. It is worth mentioning that MixUp-PME even outperforms *Oracle* on several classes. In contrast, FixMatch, as the representative method of semi-supervised learning, performs much worse than *vanilla*. As discussed in Section 3.6.4, learning-based pseudo labelling methods do not apply under data scarcity situations.

3.7.2.2 Robustness Under Class Imbalance

Following the same experimental setup as Section 3.7.2.1, the robustness of the proposed method under class imbalance is further examined. This time, 10 positive cases and 90 negative cases are selected for each class, leading to a lower intra-class positive-negative ratio $r = \frac{1}{9} = \frac{10}{90}$. Each experiment is run for three times with different random seeds. The numerical results (F1 scores) are presented in Table 3.7. Compared with Table 3.6, there three empirical findings. First, the performance gain between MixUp-PME and *vanilla* becomes larger. Second, MixUp benefits from class imbalance. Third, *vanilla* and FixMatch are less robust under severe class imbalance.

3.7.2.3 Sensitivity to α

To investigate the sensitivity of the proposed method to the hyperparameter α, an ablation study is performed to evaluate the MixUp-PME under various values of α. The experimental setup follows Section 3.7.2.1. For each value of α, each experiment is repeated three times with three random seeds. The quantitative results are depicted in Figure 3.5. While the overall performance is stable under different values of α, an interesting phenomenon can be observed: each class achieves its highest performance under different values of α.

3.7.2.4 Adaptive MixUp-PME

Intuitively, a multi-label classification task can be decomposed into C different sub-tasks. It can be seen that the highest F1 score that can be achieved by MixUp-PME for each class is higher than the counterpart achieved by *vanilla*, summarized in Table 3.8. Surprisingly, AMP even outperforms *Oracle* on all four classes. This means that, depending on the learning difficulty of the sub-task, α could be adaptive to different classes to improve the overall MTL performance. Thus, a reasonable hypothesis is that *each sub-task should have an independent* α. To validate this hypothesis, I repeat the experiments of MixUp-PME with the suitable values of $\{\alpha_k\}_{k=1}^{C}$ inferred

Table 3.7: Comparison with baselines under intra-class positive-negative ratio $r = \frac{1}{9}$. In the training set, each chest X-ray image is only partially labeled for one class. The proposed MixUp-PME achieves the best overall performance. The reported numbers are F1 scores. Blue denotes the highest number in each column (other than the *Oracle*).

Model	*Infiltration*	*Effusion*	*Atelectasis*	*Nodule*	Average
vanilla	0.2644 ± 0.0036	0.0353 ± 0.0253	0.1710 ± 0.0081	0.0685 ± 0.0297	0.1348 ± 0.0088
MixUp	0.2501 ± 0.0175	0.1677 ± 0.0130	0.1637 ± 0.0294	0.0838 ± 0.0120	0.1664 ± 0.0061
FixMatch	0.1509 ± 0.0451	0.1014 ± 0.0318	0.1442 ± 0.0189	0.1192 ± 0.0572	0.1289 ± 0.0131
MixUp-PME	0.3043 ± 0.0152	0.1552 ± 0.0203	0.1578 ± 0.0026	0.1115 ± 0.0436	0.1822 ± 0.0105
Oracle	0.3026 ± 0.0181	0.1471 ± 0.0229	0.1679 ± 0.0267	0.1964 ± 0.0267	0.2035 ± 0.0050

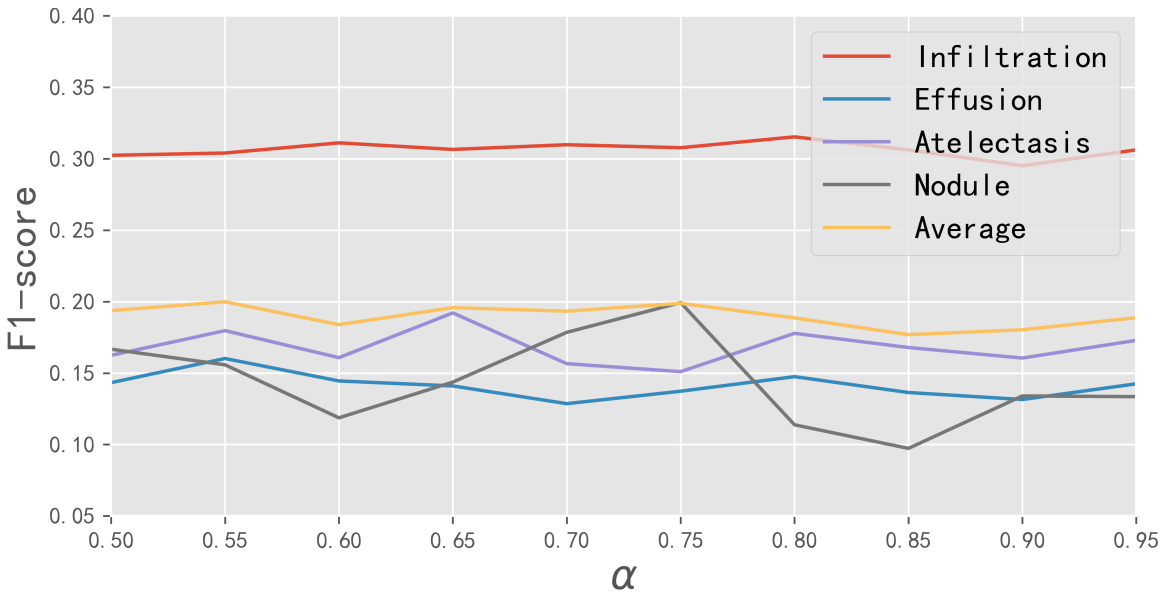


Figure 3.5: Sensitivity of MixUp-PME to α for different classes. MixUp-PME has different optimal α values for different classes.

Table 3.8: Highest performance that can be achieved by MixUp-PME for each class. The repThe last row "α" records the corresponding α value when the highest F1 scores can be achieved by MixUp-PME for each class.

	Infiltration	*Effusion*	*Atelectasis*	*Nodule*
vanilla	0.2828	0.1042	0.1759	0.1482
Oracle	0.3020	0.1305	0.1706	0.1955
MixUp-PME	0.3153	0.1602	0.1921	0.1993
α	0.75	0.55	0.65	0.70

from Figure 3.5. This adaptive design is denoted as Adaptive MixUp-PME (AMP). The results are shown in Table 3.9, where AMP outperforms *vanilla* on all four classes this time. Surprisingly, AMP even outperforms *Oracle* on three classes. It is worth mentioning that, in this work, the mean F1 score is the main performance measurement. In practice, AMP will be preferred over MixUp-PME if the performance of the individual class is more important than the average performance.

Note that the set of $\{\alpha_k\}_{k=1}^{C}$ is chosen based on posterior knowledge learned from Figure 3.5. In practice, a similar trick can be applied to find the suitable values of $\{\alpha_k\}_{k=1}^{C}$ on a small validation set. However, when C is large, this process could be troublesome. Thus, there is a trade-off between the performance and computational cost when applying AMP.

Table 3.9: AMP *vs*. MixUp-PME. In the training set, each chest X-ray image is only partially labeled for one class. AMP achieves competitive performance against MixUp-PME. The reported numbers are F1 scores. Blue denotes the highest number in each column (other than the *Oracle*)

Model	*Infiltration*	*Effusion*	*Atelectasis*	*Nodule*	Average
vanilla	0.2644 ± 0.0036	0.0353 ± 0.0253	0.1710 ± 0.0081	0.0685 ± 0.0297	0.1348 ± 0.0088
MixUp-PME	0.3024 ± 0.0159	0.1433 ± 0.0025	0.1624 ± 0.0177	0.1667 ± 0.0827	0.1937 ± 0.0147
AMP	0.3040 ± 0.0075	0.1386 ± 0.0122	0.1957 ± 0.0274	0.1253 ± 0.0040	0.1909 ± 0.0029
Oracle	0.3020 ± 0.0165	0.1305 ± 0.0202	0.1706 ± 0.0126	0.1955 ± 0.0588	0.1988 ± 0.0041

Table 3.10: Comparison between MixUp-PME and *vanilla* under multi-task learning. The reported numbers are F1 scores. Blue denotes the highest number in each column (other than the *Oracle*).

Model	*Infiltration*	*Effusion*	*Atelectasis*	*Nodule*	Average
vanilla	0.3189 ± 0.0083	0.1878 ± 0.0155	0.1729 ± 0.0225	0.0558 ± 0.0048	0.1838 ± 0.0008
MixUp-PME	0.3163 ± 0.0314	0.1785 ± 0.0060	0.1911 ± 0.0201	0.1147 ± 0.0323	0.2001 ± 0.0026
Oracle	0.3212 ± 0.0190	0.2048 ± 0.0093	0.1879 ± 0.0170	0.1332 ± 0.0491	0.2118 ± 0.0033

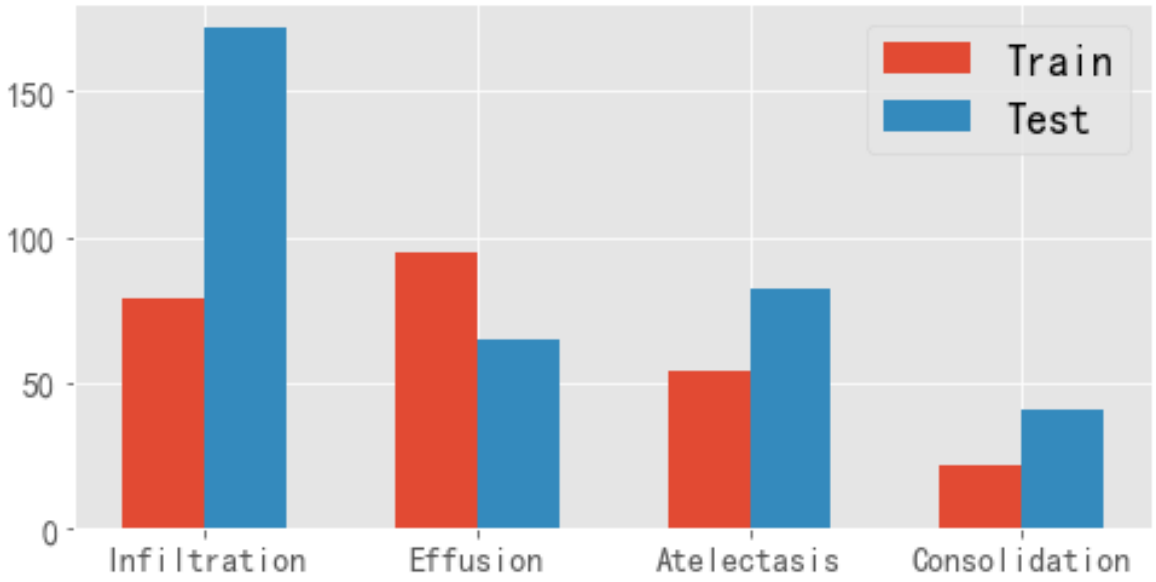


Figure 3.6: Label statistics of positive cases for the partially labeled training set and fully labeled test set.

3.7.2.5 Robustness under Multi-Task Learning

Under the setup of Section 3.7.2.1, MixUp-PME might not be able to fully leverage the advantage of MTL, which should be one of advantages of the proposed method. Thus, as an ablation study, MixUp-PME is evaluated under the more challenging situation where each chest X-ray image is labeled for more than one class. To simulate this situation, for each class of each image, a binary number is randomly generated ("0" or "1") based on a Bernoulli distribution with equal possibilities, *i.e.* Bernoulli(0.5), to indicate whether this label is available. The label distributions of the simulated partially labeled training set and the fully labeled test set are summarized in Figure 3.6. In the experiment, MixUp can not be applied anymore as it is computationally difficult to find two image-label pairs with the same set of labeled classes. Moreover, under this simulation, chest X-ray images could also be unlabeled or fully labeled. $\alpha = 0.75$ is set as the mean value of the range of α ($\alpha \in [0.5, 1)$) for MixUp-PME. The quantitative comparison between MixUp-PME and *vanilla* is presented in Table 3.10. Compared with Table 3.6, the performance of *vanilla* is negatively influenced by this challenging experimental setup. On the contrary, MixUp-PME benefits from MTL with a huge performance gain. MixUp shows robust performance by outperforming *vanilla* on all four classes by a large margin and outperforming *Oracle* on three classes.

3.8 Conclusion

As an emerging study, partially supervised multi-label classification under data scarcity is an unexplored but practical problem in partially supervised learning. This chapter serves as the first study. To tackle the issue of interest, a novel vicinal risk minimization method that is based on the principle of maximum entropy to construct the vicinity distribution is presented. The experimental results not only validate that the proposed method can be used to mitigate the data scarcity issue of partially supervised multi-label classification, but also provide insight into future research on partially supervised learning.

Chapter 4

Partially Supervised Learning with Self-Supervised Pre-Training

A common situation in partially supervised learning (PSL) is that both partially labeled data and unlabeled data are available (see Definition 2.2.3). To utilize unlabeled data, one of the possible solutions is to utilize self-supervised learning to learn transferable representations. While self-supervised pre-training on a large amount of unlabeled data followed by supervised fine-tuning on a small amount of labeled data has become a mature learning paradigm in label-efficient learning, the impact of self-supervised pre-training on PSL remains unclear. This chapter offers the first empirical study to understand the role of self-supervised pre-training in PSL and proposes a pretext task based on vicinal risk minimization (see Chapter 3).

4.1 Introduction

In addition to a small amount of partially labeled data, it is possible to have access to a large amount of unlabeled data. Medical scanners produce vast amount of image data every data, but the human resources to annotate them are rarely available. A traditional way to utilize unlabeled data is to leverage semi-supervised learning. As discussed in Chapter 3, semi-supervised learning is not a robust solution when there are only limited partial labels. Another label-efficient learning paradigm is to perform self-supervised or unsupervised pre-training. That is to say, the model of interest (*e.g.* the backbone feature extractor) is first pre-trained on the unlabeled data and then fine-tuned on partially labeled data.

First formulated in [33], self-supervised learning is a form of unsupervised learning where the learning process is not supervised by human-annotated labels. The concept of self-supervised learning originated from cognitive science by observing

how babies interact with new environments [65]. Meanwhile, representation learning aims to extract useful representations from data [14]. In this chapter, the terms "self-supervised learning", "self-supervised representation learning", "unsupervised learning", and "unsupervised representation learning" can be used interchangeably, and "self-supervised learning" is used for consistency.

Fueled by the recent success of deep learning, there is a renaissance of research on self-supervised learning. An important application of self-supervised learning is to learn transferable representations for downstream tasks, e.g. common downstream tasks for visual understanding include image recognition [152, 74], object detection [143, 109], and semantic segmentation [113, 20]. The theoretical foundation of self-supervised pre-training is *transfer learning* [131]. Concretely, given a source domain $\mathcal{D}_S$ and learning task distribution $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task distribution $\mathcal{T}_T$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$, the goal of transfer learning is to use $\mathcal{D}_S$ and $\mathcal{T}_S$ to improve the learning performance of $\mathcal{D}_T$ and $\mathcal{T}_T$. Here, $\mathcal{D}_S$ could be the unlabeled data and $\mathcal{D}_T$ could be the partially labeled data. The pre-training aims to extract transferable representations from $\mathcal{D}_S$ and $\mathcal{T}_S$. With this definition, self-supervised learning can also be understood as transfer learning from unlabeled data [139]. In the context of deep learning, the implementation of transfer learning is fairly easy: the knowledge of $\mathcal{D}_S$ and $\mathcal{T}_S$ can be represented as the weights of neural networks.

The core concept of self-supervised learning is to formulate a *pretext* task. The pretext task has its own *annotation-free* label, *i.e.* the supervision signal of the pretext task can be defined by the information within the unlabeled data. It has been reported that self-supervised learning can achieve competitive performance or even outperform standard supervised learning on image classification tasks pre-trained and evaluated on ImageNet [73, 121, 22, 21]. Though self-supervised learning has shown promising results in boosting the model performance under a supervised setup, especially when limited labels are available, its role in partially supervised learning (PSL) remains unclear. For simplicity and consistency, this chapter utilizes partially supervised multi-label classification on medical images as the downstream task to demonstrate the impact of self-supervised pre-training on PSL.

As discussed in Section 1.2, the primary research question for chapter is to leverage self-supervised learning to improve the performance of partially supervised downstream tasks. To better understand this question, I aim to answer three specific yet exploratory research questions via empirical analysis.

1. **RQ1**: Is *task affinity* a reliable tool to choose an efficient pretext task for partially supervised multi-label classification? Though task affinity has been widely adopted for supervised downstream tasks, its efficiency for PSL is still unknown.

2. **RQ2**: Does self-supervised pre-training always improve the performance of partially supervised multi-label classification? So far, there has been no study to evaluate the impact self-supervised pre-training of on partially supervised multi-label classification.

3. **RQ3**: Should multi-task learning be incorporated when designing the pretext tasks for partially supervised multi-label classification? If the answer is affirmative, multi-task learning will be a low-hanging fruit in PSL.

Before addressing **RQ1**, a formal definition of task affinity is defined in Section 4.4.1. The discussion of task affinity is bounded within the scope of machine learning. The concept arises from the study of multi-task learning [17], which aims to understand which tasks should be learned together to improve overall learning performance [169, 56]. It is also used to evaluate the relationship between the source and target tasks in transfer learning [189], *i.e.* the pretext task and the downstream task in the literature of SSL. This is a general concept and the "affinity" should be inferred from on the context. For example, in image classification, the task affinity between a dog classification task and a cat classification task should be larger than the task affinity between a dog classification task and a pneumonia classification task. Since dogs and cats share biological similarities and learning the patterns of cats can benefit learning dogs, but not learning the patterns of a thoracic disease. Similarly, the task affinity between an image classification task and a semantic segmentation task should be larger than the task affinity between an image classification task and an image reconstruction task because image classification and semantic segmentation both require visual understanding. As the task affinity between the pretext task and the downstream task can not be measured directly, I use *proxy task affinity* for quantitative comparison following SSL studies [73, 21, 23].

In MTL, grouping tasks with large task affinities can efficiently improve the training performance [56]. Thus, a larger task affinity between the pretext task and the downstream task should intuitively lead to better transfer learning performance. A few representative pretext tasks are discussed and evaluated empirically to provide insight into **RQ1**. Later in this chapter, it can be shown that, proxy task affinity is

not a robust measure for partially supervised downstream tasks, unlike its success for fully supervised downstream tasks.

For **RQ2**, self-supervised pre-training can be viewed as providing a strong initialization for the fine-tuning phase (*c.f.* random initialization). Specifically, I focus on understanding the impact of self-supervised pre-training on the downstream task under partial supervision, from a task affinity perspective. A simulated experiment is designed to investigate this question in a controllable environment. Several interesting findings are reported.

Motivated by a recent study that multi-task learning in self-supervised pre-training [37] benefits supervised downstream tasks, a reasonable hypothesis to **RQ3** is that multiple pretext tasks can also make a difference in self-supervised pre-training for PSL. To validate this hypothesis, multiple pretext tasks are designed based on proxy task affinity and evaluated to provide an empirical understanding of multi-pretext-task learning for PSL.

In addition to the above empirical studies, this work presents a pretext task based on *vicinal risk minimization* [19] that is robust for the downstream task of interest. Section 3.3.1 for details on vicinal risk minimization. Given a pair of unlabeled images and a MixUp hyperparameter [192], a vicinal image can be generated. The model of interest is pre-trained to predict the MixUp hyperparameter with two original images and the vicinal image as the input. The experimental results show that the proposed pretext task, *reverse vicinal risk minimization* can achieve competitive performance with seminal pretext tasks but with a smaller computational cost.

The main contributions of this chapter can be summarized as follows.

1. This is the first empirical study of the relationship between the pretext task and downstream task in PSL.

2. This is the first empirical study of the impact of self-supervised pre-training on partially supervised multi-label classification.

3. I explore the potential of proxy task affinity in choosing an efficient pre-training pretext task for a partially supervised downstream task and empirically show the limitations of proxy task affinity.

4. This is the first study of multi-pretext-task learning in self-supervised pre-training for partially supervised multi-label classification.

5. A pretext task based on vicinal risk minimization is proposed for PSMLC for the first time.

4.2 Related Work

4.2.1 Pretext Tasks

In the recent renaissance of self-supervised learning, the application of self-supervised learning is associated with corresponding pretext tasks. In the context of deep learning, by solving a different task that is related to the downstream task, the model of interest (a neural network) learns to extract transferable representations from the unlabeled data. These pretext tasks only utilize the information contained in the unlabeled data. However, designing a good pretext task commonly requires non-trivial human effort because *a good self-supervised task is neither simple nor ambiguous* [129]. This section discusses three families of pretext tasks.

The first family of pretext tasks utilizes the relative position information between two augmented views of the same instance. [36] uses two convolutional neural networks (CNNs) with shared weights to predict the relative positions of two patches randomly cropped from the same image. [129] generalizes the idea of [36]. [129] designs the pretext task as solving jigsaw puzzles. An image is divided into a 3×3 grid and the order of 9 patches is permuted. A CNN is then trained to predict the index of the chosen permutation. [62] randomly rotates the images by 90, 180, or 270 degrees and a CNN is trained to predicted the rotation.

The second family of pretext tasks aims to restore a corrupted instance. An image is augmented and the model is trained to reconstruct the original image given the augmented input. The main idea is that by reconstructing the image, the model (*e.g.* CNN) is able to extract transferable representations with certain level of semantic information. This idea has been widely applied in unsupervised representation learning. Popular pretext tasks include colorization [194], denoising [114], demosaicing [52], and inpainting [133]. For colorization, the original image is a colorful image. Zhang *et al.* only uses the lightness channel L of the colorful image as input to predict the corresponding a and b color channels of the same image in the CIE *Lab* colorspace [194]. For denoising, the original image is a noise-free image. A synthetic noise generated by a random distribution (commonly Gaussian) is added to the original image to create a noisy image. Then, a model is trained with noisy and noise-free image pairs in a supervised fashion to minimize a reconstruction loss between the generated images and the corresponding original noise-free images. By doing so, the model learns to map a noisy input to a clean output. Similarly, for demosaicing, the original image is in RGB domain and is subsampled into RAW domain [61]. By minimizing the reconstruction loss between the predicted RGB images

given simulated RAW input images and the corresponding original RGB images, a model is trained to learn a mapping from the RAW domain to the RGB domain. For inpainting, pixels of an original image are randomly masked to produce a corrupted image. A model is trained to recover the missing pixels given the context of unmasked pixels. This learning strategy is also known as *masked image modeling*. Recently, masked autoencoder [72] efficiently integrates this idea of reconstructing missing pixels with vision transformer [48] and achieves state-of-the-art performance in unsupervised representation learning.

In contrast to the two families mentioned above, the third family only utilizes instance-wise discrimination as the pretext task. Analogous to infant learning, a learner is only expected to identify what makes two objects similar or different, instead of directly categorizing the object of interests. This learning paradigm is also known in the literature as *contrastive learning* [9] . Contrastive learning algorithms use a neural network and a *contrastive loss* to learn a mapping from the instance space to an embedded feature space. In the feature space, the contrastive loss pulls similar instance pairs together and pushes dissimilar image pairs away. Given the same CNN backbones, contrastive learning can easily outperform other self-supervised baselines in representation learning tasks for visual understanding. A reasonable hypothesis is that instance discrimination has a larger task affinity with the downstream semantic understanding tasks (*e.g.* image classification, semantic segmentation, and object detection) than the other pretext tasks. Thus, the representations learned by contrastive learning show better generalization and transferability. In other words, contrastive learning is more efficient at extracting semantic information than the first two families of methods.

4.2.2 Contrastive Learning

Contrastive learning provides a systematic way to define the pretext task and a universal optimization strategy, where the pretext task and the optimization strategy is connected through the contrastive loss. The most popular contrastive loss is InfoNCE [130], which is based on *noise-contrastive estimation* [69]. Other forms of contrastive loss include [70, 176, 180, 77]. To construct the contrastive loss, the positive pairs and negative pairs need to be defined. For example, two random views from the same image can be considered as a positive pair and two patches from two different images can be considered as a negative pair [180]. In practice, data augmentation can be applied to the images when creating two positive views [186, 8]. The common data augmentation includes the combinations of cropping, resizing, flipping,

color distortion by grayscale conversion, color distortion by jittering, rotation, cutout, Gaussian noise, Gaussian blurring, and Sobel filtering [21]. It has been proved that minimizing InfoNCE is equivalent to maximizing a lower bound on *mutual information* between views [130]. To reduce the mutual information between views while keeping task-relevant information intact, InfoMin principle [162] is proposed to select effective views based on empirical analysis. By minimizing InfoNCE, the learner is expected to learn the invariant features shared by two views [121, 21, 162].

So far, several state-of-the-art contrastive learning methods have been proposed. InstDis [180] is an instance discrimination method related to the exemplar-based task [49] and noise-contrastive estimation [69]. The intuition behind InstDis is to learn a feature representation by discriminating individual instances, so the representation can capture apparent similarity among instances. The intuition is formulated as an instance-level non-parametric classification problem. Because of the large number of instance classes, noise-contrastive estimation is used to simplify the computation. CPC v1 [130] is designed to learn meaningful representations by predicting the future in latent feature space with autoregressive models. Along with CPC, InfoNCE is proposed. The intuition of CPC is to learn a representation which is maximally useful to predict future samples and to make the model tractable by using negative sampling. CPC v2 [75] improves the architecture and training methodology of CPC v1. CMC [161] extends the idea of CPC v1 to a multi-view setting based on the hypothesis that a powerful representation should be view-invariant. Given a set of sensory views, CMC is expected to learn a compact representation that maximizes MI between different views of the same scene. MoCo [73] proposes to use memory bank with momentum update, a dynamic dictionary, to facilitate contrastive learning. The memory bank is a large and consistent yet on-the-fly dictionary with a queue and a moving-averaged encoder driven by a momentum update to encode the keys. Based on MoCo, PIRL [121] further incorporates pretext tasks with memory bank to learn pretext-invariant representations which retain semantic information. SimCLR [21] introduces a parametric nonlinear transformation between the representation and the contrastive loss to learn a better representation. SimCLR also empirically shows that contrastive learning benefits from larger batch sizes and more training steps.

Recent studies show that negative samples are not necessary in contrastive learning [67, 23]. For example, BYOL [67] first shows that only maximizing the agreement between two augmented views of the same instance can outperform traditional contrastive learning method. SimSiam [23] further simplifies the training procedure of BYOL.

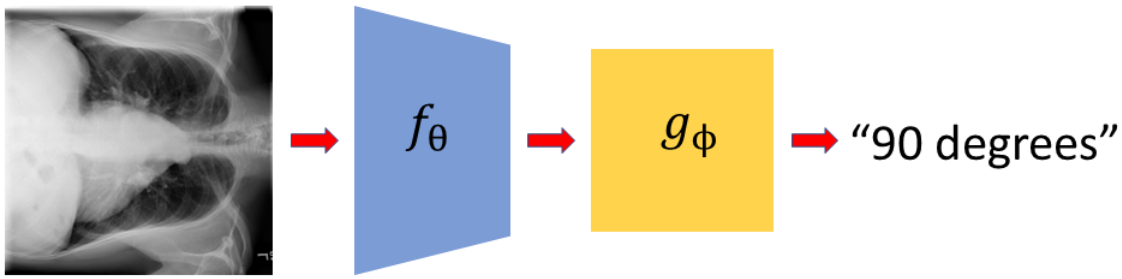


Figure 4.1: Pretext task of rotation prediction (clockwise). The network is trained to predict the rotation degree.

4.3 Preliminaries

This section provides the necessary methodological background of self-supervised pre-training for this chapter. Three representative pretext tasks for semantic understanding tasks are presented.

In this chapter, let f_θ denote the feature extractor, or the model of interest, with parameter θ and g_ϕ^t denote the auxiliary prediction head for the pretext task t with parameter ϕ. In the fine-tuning phase, only f_θ will be transferred, *i.e.* to be fine-tuned in conjunction with a new prediction head for the downstream task.

Assume there are two training datasets, a large-scale unlabeled dataset $\mathcal{S}_U$ and a small-scale partially labeled dataset $\mathcal{S}_P$, where $\mathcal{S}_P$ follows the Definition 2.2.4. In fact, $\mathcal{S}_U \cup \mathcal{S}_P$ is just an instance of Definition 2.2.3. The goal of self-supervised pre-training is to leverage $\mathcal{S}_U$ to learn transferable representations for the partially supervised downstream task fine-tuned on $\mathcal{S}_P$.

4.3.1 Rotation Prediction

Let x be an image and $\tau(\cdot)$ denote a random rotation augmentation applied on x, where the rotation degree can only be chosen from $\{0, 90, 180, 270\}$ with equal probability. The pretext task is then a multi-class classification task:

$$g_\phi^{rot} \circ f_\theta : \tau(x) \mapsto \mathbb{R}^4,$$

See Figure 4.1 for visual illustration.

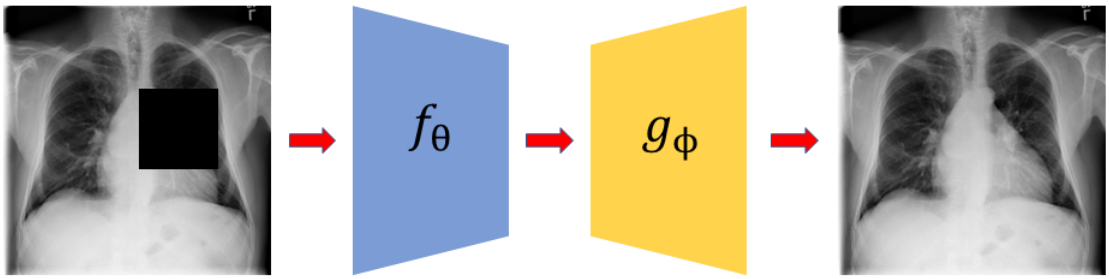


Figure 4.2: Pretext task of masked image modeling. The network is trained to reconstruct the masked image.

4.3.2 Masked Image Modeling

Let x be an image and $\tau(\cdot)$ denote a random masking operation applied on x. The optimization goal for the pretext task is to minimize a reconstruction loss, *e.g.* a mean squared loss:

$$\mathcal{L}_{MIM} = \|g_\phi^{mim} \circ f_\theta(\tau(x)) - x\|^2.$$

See Figure 4.2 for visual illustration.

4.3.3 Instance Discrimination

There are various implementations for instance discrimination, as shown in Section 4.2.2. SimSiam [23] is chosen to illustrate the concept of instance discrimination for its simplicity and robustness.

Let $x_1 = \tau(x)$ and $x_2 = \tau'(x)$ be two augmented views of the image x. The pretext task is to maximize the agreement between the embeddings of x_1 and x_2. Let $d(\cdot, \cdot)$ be a distance measure, the optimization goal is to minimize

$$\mathcal{L}_{ins} = d(g_\phi^{pred} \circ f_\theta(x_1), f_\theta(x_2)),$$

where g_ϕ^{pred} is a prediction head. The asymmetric design of g_ϕ^{pred} is designed to avoide collapsing solutions [67]. See Figure 4.3 for visual illustration.

A common choice of $d(\cdot, \cdot)$ is negative cosine similarity [172]:

$$d(g_\phi^{pred} \circ f_\theta(x_1), f_\theta(x_2)) = -\frac{g_\phi^{pred} \circ f_\theta(x_1)}{\|g_\phi^{pred} \circ f_\theta(x_1)\|} \cdot \frac{f_\theta(x_2))}{\|f_\theta(x_2))\|},$$

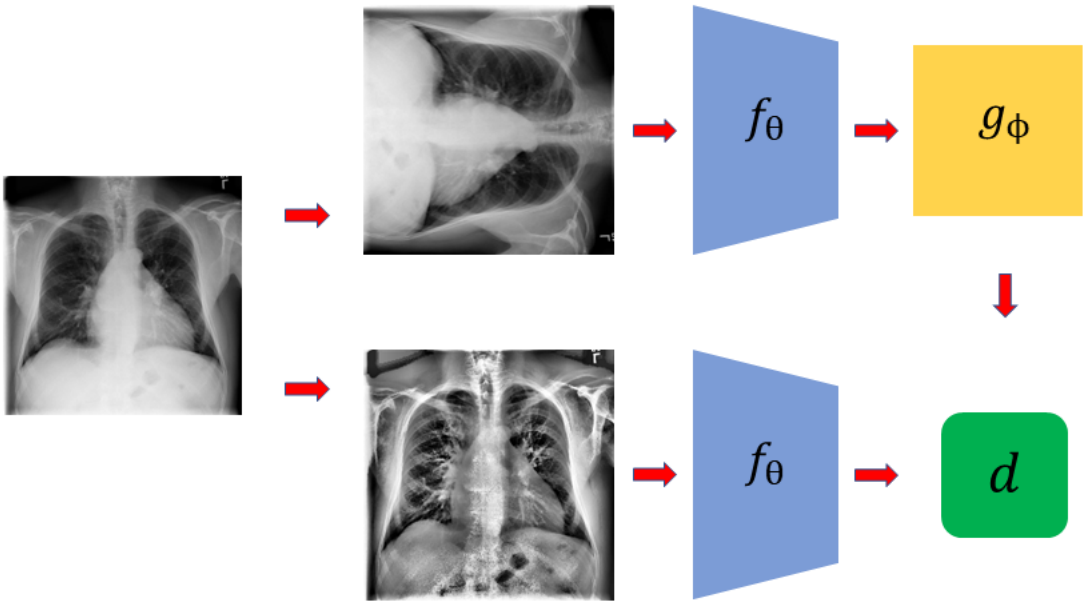


Figure 4.3: Pretext task of instance discrimination. The network is trained to learn the invariance between two augmented views of the same image.

where $\|\cdot\|$ is the norm operator. The negative cosine similarity has shown advantages in computation (the dot product operation can be efficiently implemented in GPU) and convergence [23].

4.4 Task Affinity in Transfer Learning

In machine learning, proxy task affinity has been widely adopted in multi-task learning and self-supervised learning as quantitative tools. One of the goals of this chapter is to understand whether proxy task affinity is a reliable tool to study the relationship between the pretext task and the partially supervised downstream task. This section presents the following definitions and hypotheses.

4.4.1 Task Affinity

Unlike statistical distances (*e.g.* Euclidean distance, Hausdorff distance, *etc.*), there is no such mathematical tool that can directly measure task affinity. In transfer learning, a high task affinity between the pretext task and the supervised downstream task is a positive signal when selecting the pretext task. Thus, from the perspective of optimization, if a pretext task t_p has a high task affinity with the downstream task t_d, t_p should facilitate the optimization of t_d. Concretely, at a specific time stamp

in gradient descent, a batch $\mathcal{B}$, and a loss measure $\mathcal{L}_{\mathcal{B}}$, the task affinity should be measured by

$$d_{ta} = \text{sim}(\nabla_{\theta_p}\mathcal{L}_{\mathcal{B}}, \nabla_{\theta_t}\mathcal{L}_{\mathcal{B}}), \tag{4.1}$$

where $\text{sim}(\cdot,\cdot)$ is a hypothetical similarity measure between two sets of gradients with respect to θ_p and θ_t. However, Equation (4.1) is difficult to implement in practice for three reasons. First, the training of t_p and t_d are commonly disjoint in current label-efficient learning pipeline. Second, while using a random batch $\mathcal{B}$ introduces randomness, using $\mathcal{S}_P$ as an extreme large $\mathcal{B}$ cause non-trivial computational cost. Third, Equation (4.1) is time-specific. How to choose the right time stamp in the optimization process remains an open question.

An intuitive explanation of task affinity is diagrammed in Fig. 4.4. Hypothetically, a pretext task that has no effect on the downstream task should have a zero score. In practice, random initialization could be considered as a null pretext task with zero score.

4.4.2 Proxy Evaluation

A practical challenge in defining task affinity is that task affinity can not be quantified directly. This is a common challenge in machine learning. Instead, machine learning researchers solve this issue by defining a *proxy* term. For example, [13] defines proxy $\mathcal{A}$-distance to measure $\mathcal{A}$-distance. The convergence rate of loss is used to measure the learning difficulty among different tasks. In unsupervised representation learning, the performance of clustering (*e.g.* K-means) on the representations can be used to access the quality of the learned representations indirectly.

Following the logic in Sec. 4.4.1, a proxy evaluation strategy inspired by [73, 21, 23] is used to measure task affinity between the pretext task and the downstream task[1]. Concretely, f_θ is pre-trained by the pretext task only and frozen for later fine-tuning. In the fine-tuning phase, only g_ϕ is fine-tuned for the downstream task. The performance on the downstream task is used to reflect the task affinity between two tasks: a higher performance indicates a higher task affinity score and vice versa. In this way, f_θ and g_ϕ are optimized separately, and the downstream task will not "leak" information to f_θ. Otherwise, the downstream task performance might be dominated by the downstream task in the joint optimization process. Note, when the

[1]It is worth mentioning that proxy task affinity is also called linear classification protocol in self-supervised learning, where the downstream task is fixed to be a linear classification task with fully labeled data.

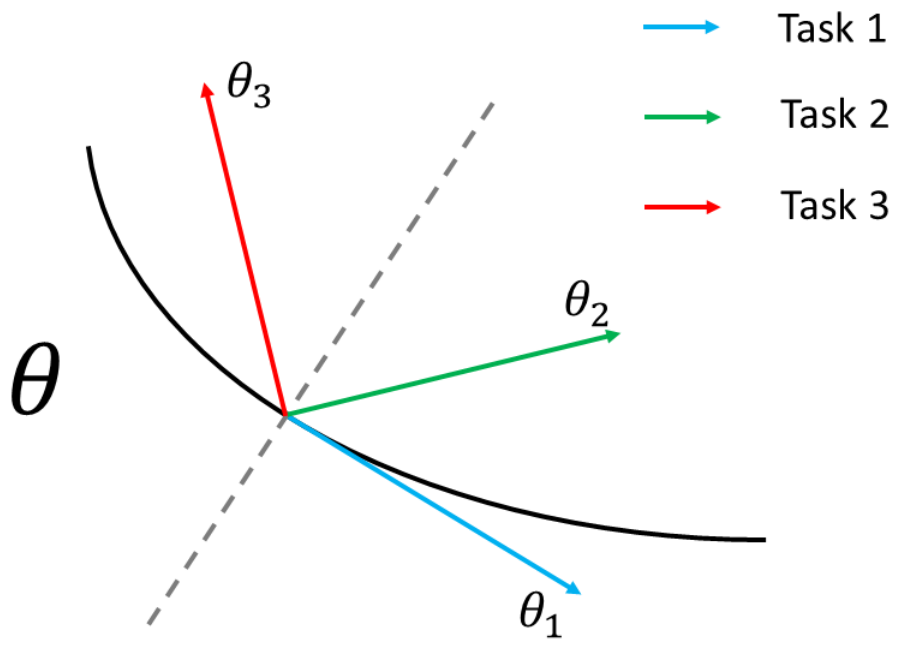


Figure 4.4: Illustration of the concept of task affinity in optimization. Intuitively, when three tasks denote three unit vectors in linear algebra, the dot product between any two tasks can denote a hypothetical task affinity score between them. Let Task 1 (the blue arrowed line tangent to the arc) be the anchor task (*i.e.* the downstream task). The ideal pretext task should optimize θ along with the direction of Task 1. Let the gray dashed line (perpendicular to Task 1) denote the zero task affinity score. Task 2 should have a positive task affinity score with Task 1 as the gradient is still in a direction that loss can be decreased. Task 3 should have a negative task affinity score with Task 1 as the gradient direction increases the loss. However, in practice, there is no such quantitative measurement.

downstream task has fully labeled data, proxy task affinity has been a benchmark tool to compare the efficiency of pretext tasks [73, 21, 23].

Mathematically, let $\mathcal{V}(t_p, t_d)$ denote the downstream task performance acquired with the pretext task t_p, the downstream task t_d, and the training procedure described above. In practice, random initialization could be considered a null pretext task with zero impact on the downstream task. Thus, let $\mathcal{V}(\texttt{RandIni}, t_d)$ denote the downstream task performance without any pretext task, where `RandIni` denotes random initialization. To ensure fairness, f_θ should be randomly initialized with the same random seed for both t_p and `RandIni`. Thus, the *proxy task affinity* is defined as below.

Definition 4.4.1 (Proxy Task Affinity)**.** Given the pretext task t_p and the downstream task t_d, the proxy task affinity under hyperparameters $*$ is

$$d_{pta}(t_p, t_d|*) = \mathcal{V}(t_p, t_d) - \mathcal{V}(\texttt{RandIni}, t_d).$$

In practice, the hyperparameters could include f_θ, g_ϕ, optimization hyperparameters, *etc*. More details can be found in Section 4.7.2.5.

4.4.3 Hypotheses of Proxy Task Affinity

With the proxy task affinity defined in Section 4.4.2, it is time to think about the role of proxy task affinity in self-supervised pre-training for partially supervised downstream tasks. Based on the prior knowledge from multi-task learning and self-supervised learning, it is natural to come up with the following two hypotheses on proxy task affinity.

Hypothesis 4.4.1. Given a pretext task and a partially supervised downstream task, the sign of proxy task affinity is independent of the number of pre-training epochs.

Hypothesis 4.4.2. Given a downstream task of interest, a pretext task with higher proxy task affinity leads to better downstream task performance.

The understanding of the above two hypotheses should substantially benefit the study of SSL and PSL. For the first time, the initial answers to these hypotheses are provided in Section 4.8 by experiments. Interestingly, both hypotheses are negated. The detailed discussion on these two hypotheses is provided in Section 4.8.

4.5 Multi-Pretext-Task Learning

Doersch *et al*. [37] suggests that incorporating multi-task learning in self-supervised pre-training can *always* improve the performance under standard supervised training setup. Dong *et al*. [40] has further shown that solving multiple pretext tasks can mitigate the *task misalignment* between the pretext and downstream tasks. However, the effect of solving multiple pretext tasks simultaneously on PSL remains unclear. Follow previous studies, it is reasonable to hypothesize that given partially supervised multi-label classification as the downstream task of interest, solving multiple pretext tasks can improve the downstream task performance.

Hypothesis 4.5.1. Given a downstream partially supervised task, solving multiple pretext tasks can improve the downstream task performance.

Concretely, let (x^t, y^t) denote the pair of input and ground truth labels for the pretext task $t \in \mathcal{T}_{pretext}$. The optimization goal of *multi-pretext-task learning* can be defined as

$$\mathcal{L}_{MPTL} = \sum_{t \in \mathcal{T}_{pretext}} w_t \mathcal{L}_t(g_\phi^t \circ f_\theta(x^t), y^t), \tag{4.2}$$

where $\mathcal{L}_t$ is the loss term for the pretext task t and w_t is the corresponding weight with $\sum_{t \in \mathcal{T}_{pretext}} w_t = 1$. Note, in Equation (4.2), only f_θ is shared across pretext tasks and each pretext task has its own task-dependent prediction head.

4.6 Reverse Vicinal Risk Minimization

[45] has shown that vicinal risk minimization [192] can benefit the partially supervised learning on multi-object images. This section further proposes a pretext task based on vicinal risk minimization for partially supervised multi-label classification. In MixUp [192], the vicinal image $\tilde{x}$ is first generated by two randomly selected unlabeled images x_i and x_j by a linear combination, as shown in Equation (4.3), where $\lambda \sim \text{Beta}(\alpha, \alpha)$ is a MixUp hyperparameter.

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \tag{4.3}$$

The corresponding binary labels y_i and y_j are linearly added in the same way to generate a vicinal label.

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j \tag{4.4}$$

For vicinal risk minimization, the generated vicinal image and label pair are directly used in supervised learning to improve model generalization.

In contrast to vicinal risk minimization, I reverse the MixUp process and design the pretext task as predicting the MixUp hyperparameter λ. Let the feature extractor is f_θ and the prediction head is g_ϕ^{rvrm}. The optimization goal for the pretext task is then a regression task:

$$\mathcal{L}_{VRM} = \|g_\phi^{rvrm} \circ f_\theta(\boldsymbol{x}) - \lambda\|^2, \tag{4.5}$$

where $\boldsymbol{x}$ is simply a concatenation of $(x_i, x_j, \tilde{x})$ alongside the channel dimension. In the inference phase, given an input image x_k, $\boldsymbol{x}$ is then the concatenation of (x_k, x_k, x_k). This actually fits the seminal network architectures that take three-channel RGB images as input, *e.g.* VGG [152], ResNet [74], and DenseNet [81].

Note, in the whole training and inference process, there is no use of ground truth label y or vicinal label $\tilde{y}$. Thus, this is a self-supervised task. I denote this pretext task as *reverse vicinal risk minimization* (RVRM). In comparison with three pretext tasks in Section 4.3, RVRM requires fewer computational costs than the seminal baselines but with comparable or even better performance, which will be discussed in Section 4.8.4.

The pretext task is illustrated in Figure 4.5.

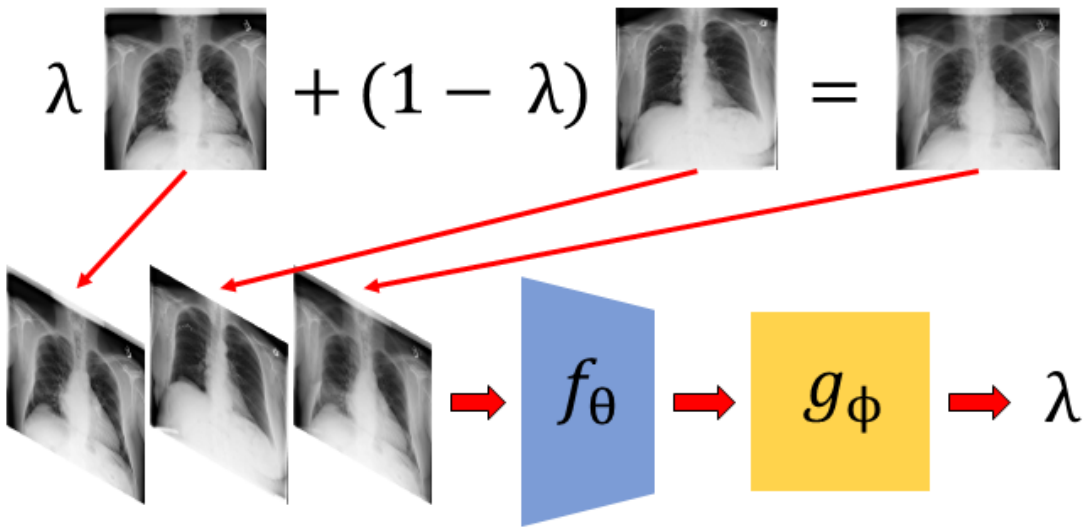


Figure 4.5: Pretext task based on reverse vicinal risk minimization. Given two randomly sampled images and the generated vicinal images, the network is trained to predict the MixUp hyperparameter λ.

4.7 Experimental Setup

The purpose of experimental design is threefold. First, I aim to understand the relationship between the pretext and downstream tasks. Second, the first empirical study of the impact of self-supervised pre-training on partially supervised multi-label classification is provided. Third, the hypotheses proposed in Section 4.4 are discussed. It should be highlighted that the experiments in this section are not designed to beat the state-of-the-art self-supervised learning baselines. Instead, the numerical results of the simulated experiments provide empirical insight.

4.7.1 Data

The public multi-label dataset of thoracic conditions ChestX-ray14 [177] is used in this study and its default batch splits are adopted to ensure reproducibility. The second batch is selected as the pre-training set, which contains 10^4 unlabeled chest X-ray images. The eleven most common diseases are selected as the tasks of interest, which are *infiltration*, *effusion*, *atelectasis*, *nodule*, *consolidation*, *pneumothorax*, *cardiomegaly*, *fibrosis*, *pleural thickening*, *mass*, and *emphysema*. From the fifth batch, 200 cases are sampled for each of eleven diseases to form a training set consisting of 2200 chest X-rays images in total. For simplicity, each disease has 100 positive cases and 100 negative cases to ensure both intra-class and inter-class class balance in terms of number of examples. To simulate the partial labels, the sampled 200 chest X-ray images for each disease are only labeled for this diseases, *i.e.* the rest of ten diseases are unlabeled. Similarly, another 2200 chest X-rays images are sampled for eleven diseases to form a balanced test set.

4.7.2 Implementation

Following the setup of [140], DenseNet121 [81] is chosen as the feature extractor (f_θ). The weighted loss in Equation (3.4) is minimized by a standard Adam optimizer [97] with batch size 64. In the pre-training stage, the learning rate is fixed to be 10^{-2} and in the fine-tuning stage, the learning rate is fixed to be 10^{-2}. In the inference phase, 0.5 is set as the default threshold for the predicted probability score.

In this chapter, I focus on the overall learning performance of self-supervised learning methods. Thus, the evaluation metric is AUROC (see Section 2.3.2.2 for details of AUROC and see Section 2.3.2.3 for the reasons). For each class of interest, AUROC is computed. Then, the mean of eleven AUROCs is used to measure the overall learning performance, denoted as mAUROC.

All experiments are conducted in PyTorch [132] on an NVIDIA Tesla V100. All chest X-ray images are resized to a fixed size of 224×224. As a pre-processing step, instance normalization [30] in Equation (3.6) is performed on each chest X-ray image.

4.7.2.1 Rotation Prediction

The original image is rotated by $\{90 \times i\}$ degrees where i is an integer randomly sampled from a discrete uniform distribution $\{0, 1, 2, 3\}$. The auxiliary prediction head g_ϕ^{rot} is just the last layer of DenseNet121, which is a fully-connected layer with 1024 input channels and 4 output channels.

4.7.2.2 Masked Image Modeling

A 4×4 grid is first generated to split each input image into 16 patches. A patch is randomly selected out of 16 patches for each image. This patch is masked out by replacing original pixels values with zeros. The model $g_\phi^{mim} \circ f_\theta$ is implemented as a FCN [113], where the auxiliary prediction head g_ϕ^{mim} is a 2D transposed convolutional layer with kernel size 64, stride size 32, dilation rate 1, and padding size 16.

4.7.2.3 Instance Discrimination

The pretext task with negative cosine similarity is implemented as SimSiam [23]. The 3-layer projector has the following architecture: $FC(1024, 1024) \rightarrow BN \rightarrow ReLU \rightarrow FC(1024, 1024) \rightarrow BN \rightarrow ReLU \rightarrow FC(1024, 2048) \rightarrow BN$, where $FC(a, b)$ denotes a fully-connected layer with a input channels and b output channels, BN denotes batch normalization [83], and ReLU denotes rectified linear unit [126]. The 2-layer predictor has the following architecture: $FC(2048, 1024) \rightarrow BN \rightarrow ReLU \rightarrow FC(1024, 2048)$.

4.7.2.4 Data Augmentation

Previous studies on contrastive learning tend to focus on RGB images [21, 10]. Thus, a direct application of data augmentation policy adopted by these studies is not feasible as medical images are commonly grayscale images. To overcome this issue, I adapt the data augmentation policy for single-channel images, which is illustrated in Listing 4.1.

```
1 import random
2 import torchvision.transforms as transforms
3 from PIL import ImageFilter
4
5 class GaussianBlur(object):
```

```
    """Gaussian blur augmentation"""

    def __init__(self, sigma=[0.1, 2.]):
        self.sigma = sigma

    def __call__(self, x):
        sigma = random.uniform(self.sigma[0], self.sigma
    [1])
        x = x.filter(ImageFilter.GaussianBlur(radius=sigma
    ))
        return x

augmentation = [
    transforms.RandomResizedCrop(224, scale=(0.2, 1.)),
    transforms.Grayscale(num_output_channels=3),
    transforms.RandomApply([
        transforms.ColorJitter(0.4, 0.4, 0.4, 0.1)
        ], p=0.8),
    transforms.RandomGrayscale(p=0.2),
    transforms.RandomApply([
        GaussianBlur([0.1, 2.])
        ], p=0.5),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    ]
```

Listing 4.1: PyTorch code for the data augmentation policy for instance discrimination.

4.7.2.5 Proxy Task Affinity

I implement the proxy task affinity based on linear classification protocol [73, 21], which has been widely used to evaluate the quality of the representations learned by self-supervised learning methods. Given a network backbone f_θ, *e.g.* a DenseNet121 without the last layer, the weights of f_θ are first frozen and a randomly initialized fully-connected layer (*i.e.* g_ϕ) is appended after f_θ as the classification head. In the training phase, only the weights of the classification head are updated and the weights of f_θ are fixed. Then, the classification performance (mAUROC) on the test set (*i.e.* $\mathcal{V}$ in Definition 4.4.1) is used as the proxy evaluation for the learned representations.

4.7.2.6 Multi-Pretext-Task Learning

As discussed in Section 4.5, the DenseNet121 backbone is shared across different pretext tasks and each pretext task has its own task-dependent prediction head. Equal weights are adopted for losses in Equation (4.2). The choice of equal weights is further discussed in Section 4.8.3.

4.7.2.7 Vicinal Risk Minimization

The implementation of vicinal risk minimization follows MixUp, which can be found at `https://github.com/facebookresearch/mixup-cifar10`. The value 1 is utilized for both shape parameters of the Beta distribution from which the MixUp parameters are sampled (see Section 4.6).

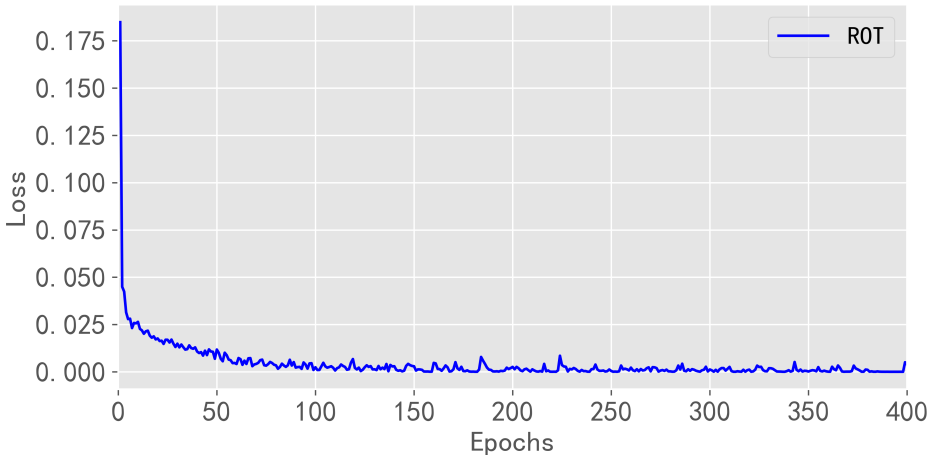
4.8 Empirical Analysis

4.8.1 Relationship between Pretext and Downstream Tasks

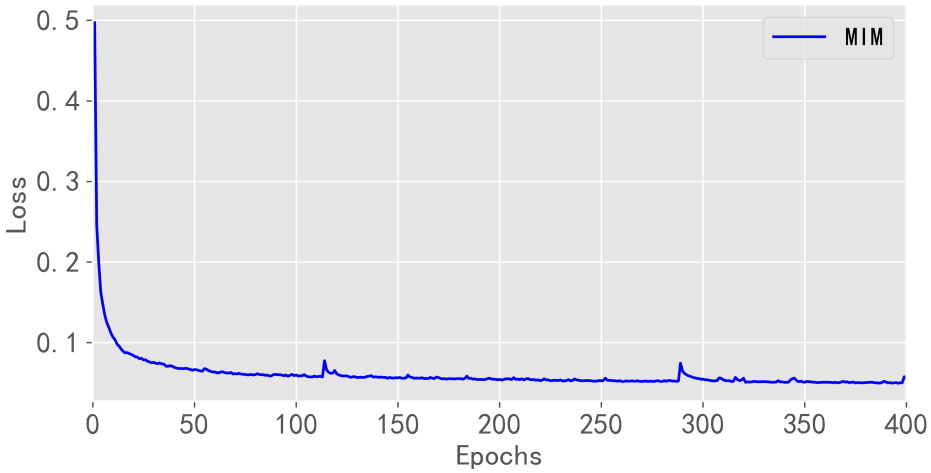
Three pretext tasks introduced in Section 4.3 are considered in the linear classification protocol, which are rotation prediction (`ROT`), masked image modeling (`MIM`), and instance discrimination (`InstDis`). As the feature extractor is fixed during the linear classification protocol, only the prediction head for the downstream task (partially supervised multi-label classification) is fine-tuned. The mAUROC over eleven classes is used as the proxy evaluation performance to measure the task affinity between the pretext and downstream tasks.

Three pretext tasks are pre-trained for 400 epochs. The learning curves (*i.e.* the loss for each pretext task) are presented in Figure 4.6, where all losses converge. In the mean time, the corresponding learning performance under the linear classification protocol is also depicted in Figure 4.7. The reported number is the mean of mAUROCs over three random seeds.

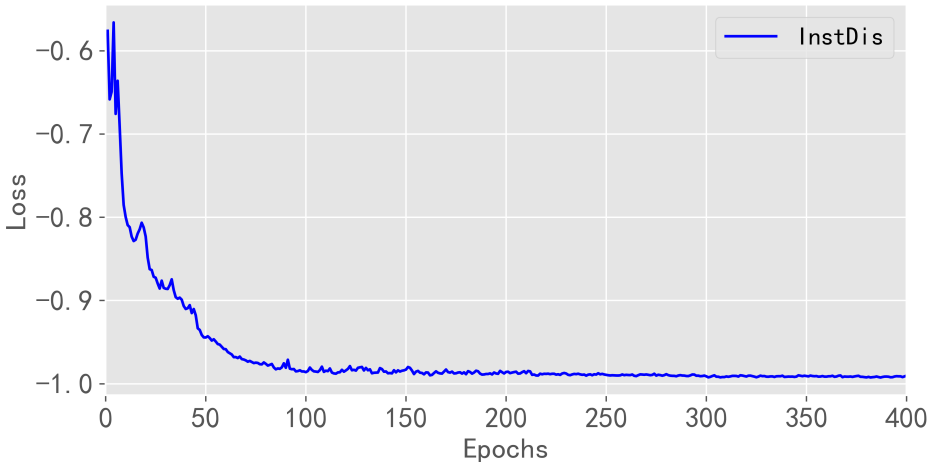
There are three important findings. First, the linear classification performance suggests that `MIM` has a larger proxy task affinity with partially supervised multi-label classification than `ROT` and `InstDis` with the current experimental setup. Second, the proxy task affinity for a pretext task can be positive or negative given a downstream task. `ROT` and `InstDis` can achieve lower linear classification performance than `RandIni` (no pre-training) with insufficient pre-training or too much pre-training. This suggests that the proxy task affinity is dependent on the number of training epochs, which actually invalidates Hypothesis 4.4.1. Last but not least, a



(a) ROT



(b) MIM



(c) InstDis

Figure 4.6: Learning curves for three different pretext tasks. The losses for all pretext tasks converge.

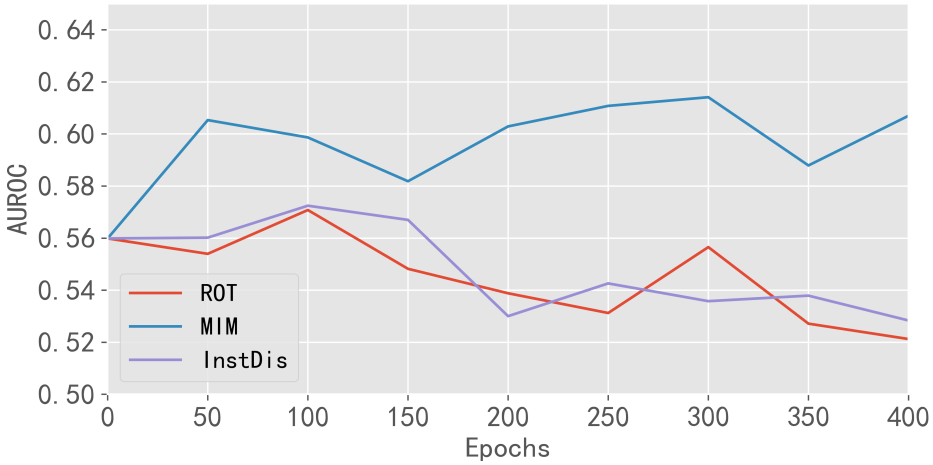


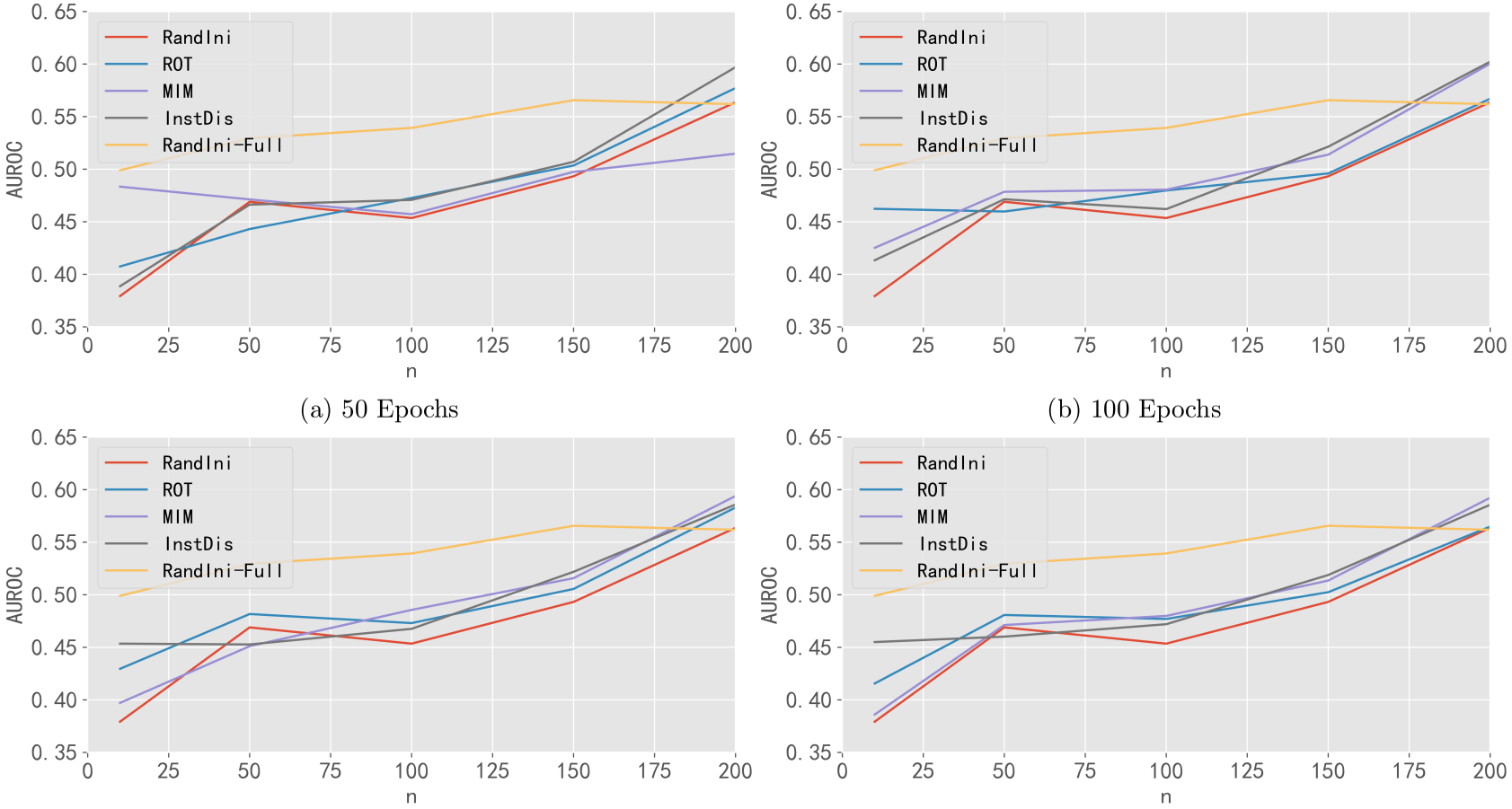
Figure 4.7: Linear classification performance for three different pretext tasks. The reported AUROCs are the mean mAUROCs over three random seeds.

larger number of pre-training epochs does not necessarily improve the performance, in contrast to the finding in self-supervised learning [73, 21].

4.8.2 Impact of Self-Supervised Pre-Training

To better understand the impact of self-supervised pre-training on partially supervised multi-label classification, the experiments in Section 4.8.1 are repeated without linear classification protocol. That is to say, the feature extractor backbone is fine-tuned jointly with the prediction head. In this section, two control variables are the number of partially labeled instances for each class and the number of pre-training epochs. Same as the previous section, mAUROCs under three random seeds are considered as the downstream task performance for each pretext task. The reported numbers are depicted in Figure 4.8 and Figure 4.9, respectively. In addition to `RandIni`, I include a randomly initialized baseline fine-tuned with full labels, denoted as `RandIni-Full`.

There are four important findings. First, a higher task affinity does not always lead to a higher downstream task performance, which invalidates Hypothesis 4.4.2. A good example is `MIM`, which has the highest task affinity among three pretext tasks (Figure 4.7). `MIM` tends to have lower performance than the other two pretext tasks with small n. Similarly, even with a lower task affinity than `MIM`, `InstDis` can achieve competitive or even better results. Second, PSL baselines can even outperform the supervised learning baseline with full labels (`RandIni-Full`) in Figure 4.9d. This is an interesting phenomenon because a supervised learning baseline with full labels is commonly deemed as *Oracle* in the literature of PSL [39, 45], which is an upper bound for the downstream task performance. A reasonable explanation is that full



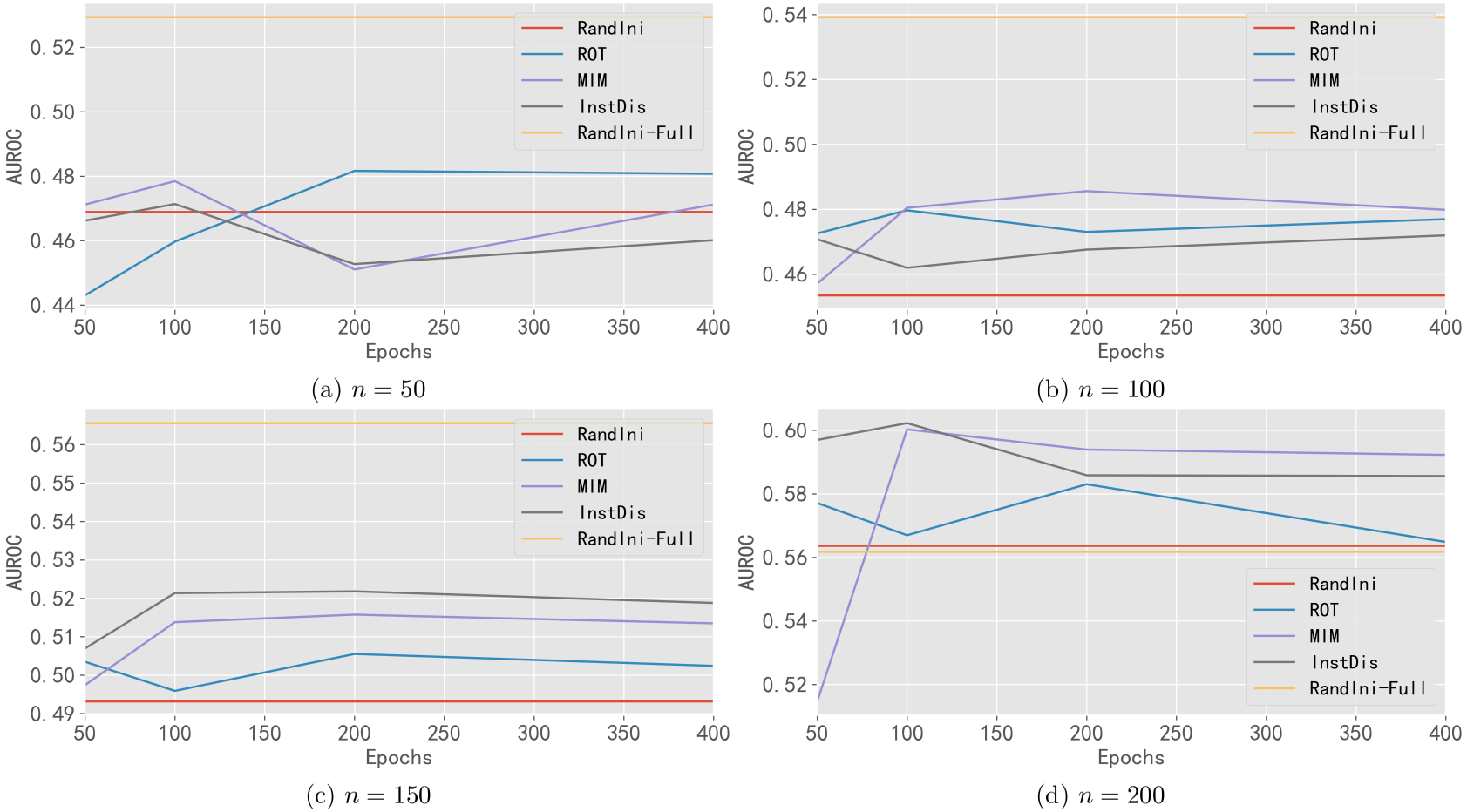
(a) 50 Epochs

(b) 100 Epochs

(c) 200 Epochs

(d) 400 Epochs

Figure 4.8: Downstream task performance for different values of n, the number of partially labeled instances for each class given specific pre-training epochs. The reported AUROCs are the mean mAUROCs over three random seeds.



(a) $n = 50$

(b) $n = 100$

(c) $n = 150$

(d) $n = 200$

Figure 4.9: Downstream task performance with different numbers of pre-training epochs given specific numbers of partially labeled instances for each class (n). The reported AUROCs are the mean mAUROCs over three random seeds.

Table 4.1: Class-wise downstream task performance. The reported numbers are the mean and standard deviation of AUROCs over three random seeds. Blue denotes the highest AUROC for each class (each row).

	Model				
	`RandIni`	`ROT`	`MIM`	`InstDis`	`RandIni-Full`
Infiltration	0.563 ± 0.012	0.568 ± 0.017	0.597 ± 0.008	0.603 ± 0.022	0.580 ± 0.054
Effusion	0.563 ± 0.017	0.577 ± 0.008	0.623 ± 0.008	0.650 ± 0.021	0.632 ± 0.082
Atelectasis	0.532 ± 0.002	0.602 ± 0.028	0.602 ± 0.012	0.585 ± 0.047	0.605 ± 0.043
Nodule	0.567 ± 0.031	0.583 ± 0.018	0.627 ± 0.010	0.598 ± 0.033	0.508 ± 0.012
Consolidation	0.570 ± 0.011	0.632 ± 0.020	0.647 ± 0.014	0.665 ± 0.036	0.548 ± 0.034
Pneumothorax	0.578 ± 0.042	0.628 ± 0.012	0.620 ± 0.012	0.595 ± 0.007	0.532 ± 0.025
Cardiomegaly	0.598 ± 0.029	0.602 ± 0.006	0.625 ± 0.004	0.628 ± 0.040	0.588 ± 0.035
Fibrosis	0.450 ± 0.025	0.405 ± 0.008	0.373 ± 0.008	0.438 ± 0.047	0.518 ± 0.043
Pleural Thickening	0.622 ± 0.012	0.625 ± 0.016	0.632 ± 0.012	0.628 ± 0.049	0.557 ± 0.018
Mass	0.543 ± 0.026	0.560 ± 0.018	0.610 ± 0.010	0.598 ± 0.022	0.582 ± 0.043
Emphysema	0.613 ± 0.020	0.632 ± 0.020	0.648 ± 0.014	0.635 ± 0.047	0.530 ± 0.012
Average	0.564 ± 0.016	0.583 ± 0.013	0.600 ± 0.012	0.602 ± 0.032	0.562 ± 0.030

labels introduce class imbalance. Note, the designed partial labels experiment has balanced class distribution. Though full labels have ten times more class-wise labels, most of labels belong to negative cases and common diseases have more positive cases than uncommon diseases. Third, given n, more pre-training epochs can lower the downstream task performance. This could be caused by the pre-training overfitting, *i.e.* the feature extractor overfits the pretext task and exacerbates the task misalignment. Fourth, though all three pretext tasks can improve the performance of `RandIni`, there is no "best" pretext task.

As shown in Figure 4.8 and Figure 4.9, partially supervised multi-label classification is a challenging downstream task. Three pretext tasks introduced in Section 4.3 represent three families of state-of-the-art methods for multi-class classification. In contrast to multi-class classification on iconic images, multi-label classification has multiple objects at multiple locations in the same image [45].

Unfortunately, by the time this chapter was written, there is no existing efficient pretext task that has been designed for multi-label classification or partially supervised multi-label classification. As a complement to **RQ1**, here are three general suggestions for choosing or designing pretext tasks. First, for downstream tasks with data scarcity, long pre-training should be avoided. The early stopping strategy can be adopted based on the learning curves. The best-performing pre-training epochs seem to overlap with the epochs when the loss starts to converge. Second, among the discussed three pretext tasks, `InstDis` gives the most robust performance when data scarcity is not severe in the downstream task and `MIM` gives the most robust performance under severe data scarcity. Third, `ROT` has worse overall performance than `MIM` and `InstDis`. Meanwhile, `ROT` also has lower proxy task affinity and learning difficulty (easy to converge) than `MIM` and `InstDis`. This suggests that when designing a pretext task, the proxy task affinity and learning difficulty should be taken into consideration. The proxy task affinity under small pre-training epochs and the learning difficulty should not be low.

For **RQ2**, though self-supervised pre-training has the potential to improve the performance of the downstream tasks, the choice of the pretext task, the number of pre-training epochs, and the number of partial labels in the fine-tuning stage are important factors to consider.

Within the downstream task (partially supervised multi-label classification), each sub-task (class) should also have its own learning difficulty. This section also aims to understand the impact of self-supervised pre-training on each class. Given $n = 200$, the highest downstream task performance is presented for all five baselines in

Table 4.1. Similar to the findings in the previous section, `MIM` and `InstDis` are better choices than `ROT`. `InstDis` achieves the highest AUROCs for most classes. `MIM` comes second and has the lowest standard deviations most of the time. `RandIni-Full` only significantly outperforms `MIM` and `InstDis` on *fibrosis*, a challenging disease for partially supervised multi-label classification. The results in Table 4.1 also suggest that the performance of a multi-label classification task can be improved by dropping a few negative labels (or weighted sampling) and transforming the problem into a partially supervised multi-label classification task.

4.8.3 Effect of Multi-Pretext-Task Learning

Following the discussion in Section 4.5, multiple pretext tasks are pre-trained together to understand the interactions between pretext tasks. As the first step, linear classification performance is reported under the linear classification protocol in Section 4.8.1. The results are shown in Figure 4.10. In contrast to single pretext tasks (Figure 4.7), multi-pretext-task learning (MPTL) efficiently improves the proxy task affinity and robustness. Except `ROT-MIM-INS`, `ROT-MIM`, `ROT-INS`, and `MIM-INS` all consistently outperform the `RandIni` baseline (0 pre-training epochs), which supports Hypo 4.5.1, and the overfitting phenomenon with a large number of pre-training epochs seems to be efficiently alleviated.

In addition, the downstream task performance of MPTL is presented in Figure 4.11, where the feature extractor and prediction head are fine-tuned jointly. MPTL does improve the performance (the highest mAUROC that a model of interest can achieve under a determined experimental setup) over single pretext tasks when enough partially labeled examples (n) are available. However, one should also notice that MPTL exposes disadvantages when n is small, *e.g.* `ROT-MIM-INS` with $n = 50$ in Figure 4.11a. It is interesting that `ROT-MIM-INS` does not achieve the highest performance in contrast to `ROT-MIM`, `ROT-INS`, and `MIM-INS`. This suggests *using more pretext tasks might not help*. Similarly, even though `MIM` and `INS` perform well alone, `MIM-INS`, the combination of `MIM` and `INS`, does not shows the best result. The class-wise mAUROCs for MPTL are summarized in Table 4.2 based on the best-performing epochs given $n = 200$. The empirical findings suggest that, while MPTL can improve the downstream task performance, additional attention should be paid to the size of fine-tuning dataset and choice of pretext tasks. Now, it is reasonable to provide an affirmative answer to **RQ3**. However, one should also realize the trade-off between the performance gain and additional computational cost when adopting MPTL.

Last but not least, it is worth mentioning that a brute-force grid search can find the optimal weight assignment empirically and how to find the optimal weight assignment (*e.g.* by evolutionary algorithm [16] or Bayesian optimization [156]) is beyond the scope of this thesis. Here, for simplicity, equal weights are adopted just to illustrate the effect of MPTL, *i.e.* whether MTL helps when designing a pretext task.

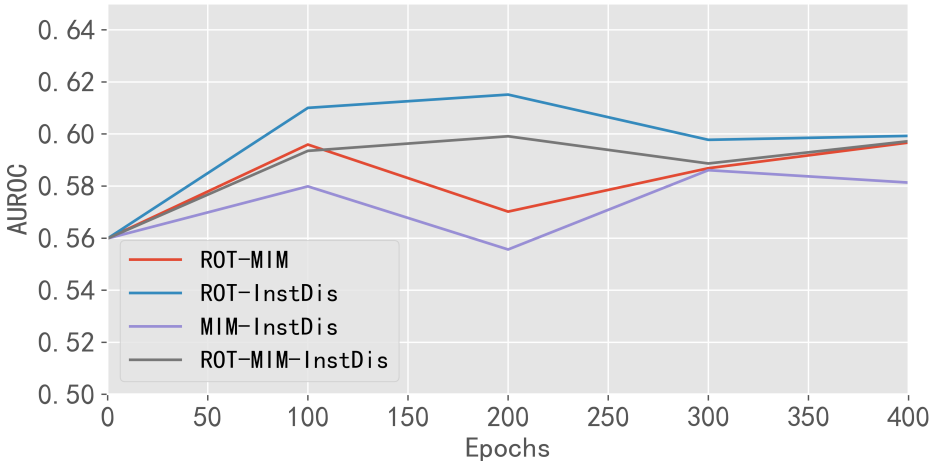
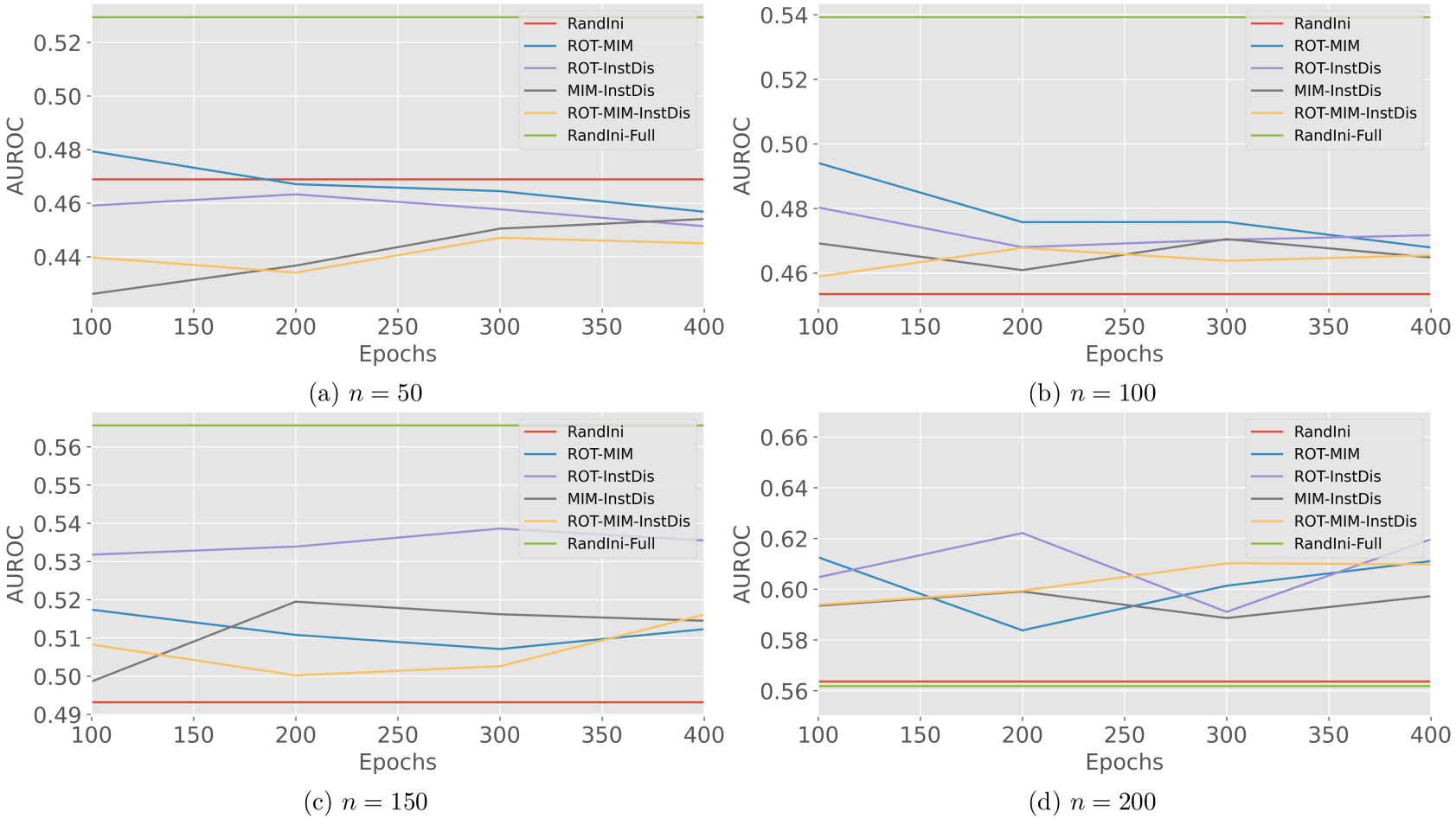


Figure 4.10: Linear classification performance of multi-pretext-task learning. The reported AUROCs are the mean mAUROCs over three random seeds.

4.8.4 Evaluation of Reverse Vicinal Risk Minimization

The pretext task is denoted as `RVRM`. The learning curve and the linear classification performance are presented in Figure 4.12 and Figure 4.13, respectively. The loss for `RVRM` does converge and `RVRM` can have positive proxy task affinity. The overall downstream task performance of `RVRM` and the class-wise performance are presented in Figure 4.14 and Table 4.2, respectively. In contrast to Figure 4.9 and Table 4.1, `RVRM` shows competitive or even superior performance with other single pretext task baselines. `RVRM` also suffers from overfitting with long pre-training. However, one should notice that, compared with `MIM` and `INS`, two best-performing baselines, `RVRM` achieves competitive performance but requires much less computational cost. Similar to [39], the computational cost of sampling and Equation (4.3) is trivial in comparison with forward and backward pass. Concretely, `RVRM` does not require expensive transposed convolution operation in `MIM` or additional memory footprint or network architecture in `INS`. Meanwhile, `RVRM` outperforms `ROT` at different scales without increasing the computational cost.



(a) $n = 50$ (b) $n = 100$

(c) $n = 150$ (d) $n = 200$

Figure 4.11: Downstream task performance of multi-pretext-task learning with different numbers of pre-training epochs given specific numbers of partially labeled instances for each class (n). The reported AUROCs are the mean mAUROCs over three random seeds.

Table 4.2: Class-wise downstream task performance for MPTL and RVRM. The reported numbers are the mean and standard deviation of AUROCs over three random seeds. Blue denotes the highest AUROC for each class among MPTL baselines (except the last column). Red denotes that RVRM outperforms the highest AUROC for each class in Table 4.1.

	Model				
	ROT-MIM	ROT-INS	MIM-INS	ROT-MIM-INS	RVRM
Infiltration	0.607 ± 0.002	0.578 ± 0.010	0.603 ± 0.014	0.603 ± 0.008	0.583 ± 0.009
Effusion	0.657 ± 0.014	0.665 ± 0.011	0.645 ± 0.031	0.631 ± 0.006	0.617 ± 0.021
Atelectasis	0.600 ± 0.008	0.625 ± 0.004	0.588 ± 0.044	0.617 ± 0.002	0.595 ± 0.014
Nodule	0.645 ± 0.011	0.643 ± 0.005	0.615 ± 0.011	0.597 ± 0.044	0.585 ± 0.015
Consolidation	0.645 ± 0.015	0.665 ± 0.015	0.650 ± 0.011	0.643 ± 0.019	0.637 ± 0.035
Pneumothorax	0.623 ± 0.002	0.630 ± 0.029	0.630 ± 0.011	0.642 ± 0.018	0.638 ± 0.023
Cardiomegaly	0.657 ± 0.024	0.648 ± 0.006	0.650 ± 0.011	0.641 ± 0.008	0.637 ± 0.002
Fibrosis	0.373 ± 0.018	0.455 ± 0.011	0.373 ± 0.002	0.400 ± 0.025	0.448 ± 0.041
Pleural Thickening	0.650 ± 0.011	0.640 ± 0.018	0.652 ± 0.012	0.646 ± 0.013	0.678 ± 0.033
Mass	0.618 ± 0.008	0.627 ± 0.002	0.580 ± 0.004	0.613 ± 0.009	0.593 ± 0.041
Emphysema	0.663 ± 0.006	0.667 ± 0.005	0.645 ± 0.015	0.677 ± 0.006	0.615 ± 0.021
Average	0.613 ± 0.007	0.622 ± 0.001	0.603 ± 0.011	0.610 ± 0.001	0.602 ± 0.016

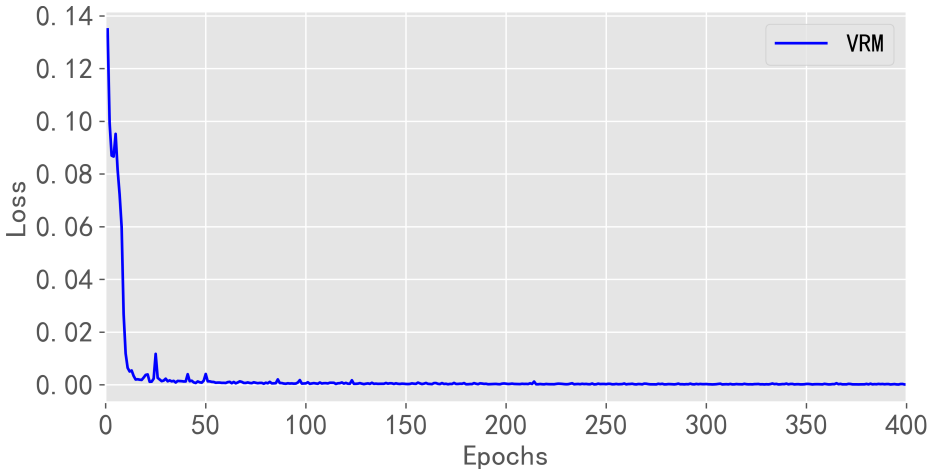


Figure 4.12: Learning curve for the pretext task based on vicinal risk minimization.

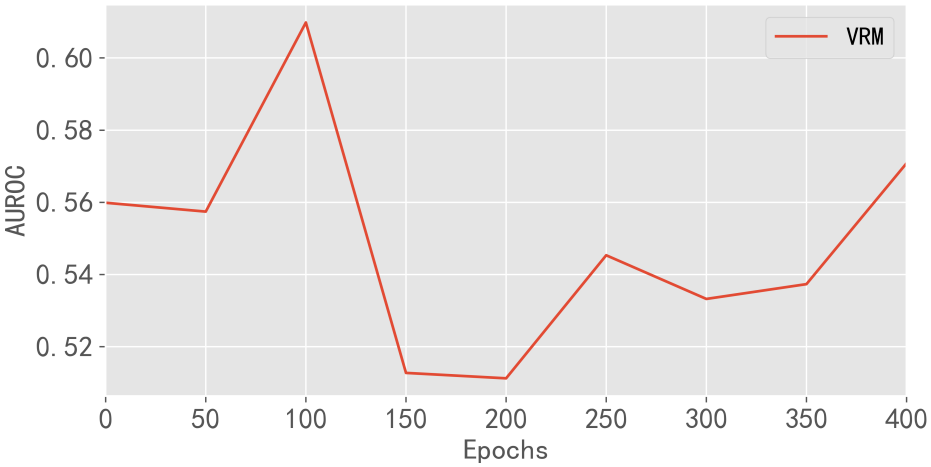


Figure 4.13: Linear classification performance of the pretext task based on vicinal risk minimization. The reported AUROCs are the mean mAUROCs over three random seeds.

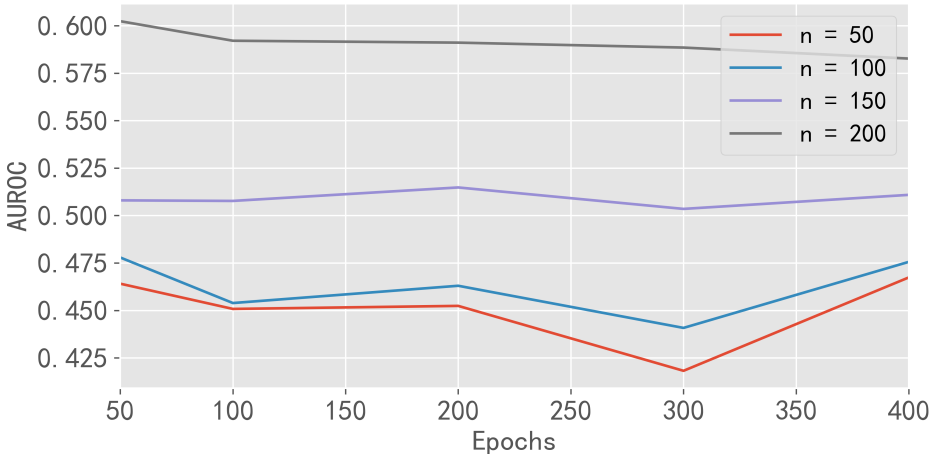


Figure 4.14: Downstream task performance of the pretext task based on vicinal risk minimization with different number of pre-training epochs given n. The reported AUROCs are the mean mAUROCs over three random seeds.

4.9 Conclusion

This chapter discusses the role of self-supervised pre-training for partially supervised multi-label classification. The empirical results show that proxy task affinity is not a reliable tool to choose an efficient pretext task when the downstream task is partially supervised. This chapter also offers the first empirical study of relationship between the pretext task and partially supervised downstream task, the first empirical study of the impact of self-supervised pre-training in partially supervised multi-label classification, and the first study of multi-pretext-task learning in self-supervised pre-training for partially supervised multi-label classification. A novel pretext task, reverse vicinal risk minimization, is proposed and evaluated. It is also worth mentioning that the numerical results in this chapter are dependent on the data scarcity setups which are designed to be challenging. The numbers are only used as a proxy measure to compare the efficiency of different pretext tasks. For practical tasks, the choice of pretext task and optimization strategy shall be discussed in a case-by-case basis.

Chapter 5

Federated Partially Supervised Learning

Data governance has played an instrumental role in securing the privacy-critical infrastructure in the medical domain and has led to an increased need of *federated learning*. While Chapter 3 and Chapter 4 assume a centralized training environment for *partially supervised learning*, partially labeled datasets in the sense defined in Chapter 2 can be stored separately in a federated system. This challenging situation poses an emerging research field, *federated partially supervised learning*. The federated setup commonly comes with additional and demanding requirements in terms of data communication and users' privacy. More importantly, the federated environment will make machine learning algorithms that are designed for a centralized environment inefficient or even infeasible. This chapter formulates the problem of federated partially supervised learning for the first time and offers the first empirical study of the impact of data decentralization on partially supervised learning. It also proposes a federated solution for the problem of interest.

5.1 Introduction

Existing studies in partially supervised learning (PSL) [134, 63, 199, 54, 150, 182, 193, 39, 45] tend to assume that multiple partially labeled datasets can be collected and stored at the same place. However, such a data centralization setup can be difficult to fulfill. For example, under a clinical setup, each partially labeled dataset might be owned by each hospital, and due to data governance constraints, it is impossible to transfer the data from each data owner to the model trainer directly. In the medical domain, data regulations commonly refer to situations that the data stored in the client are not allowed to be transferred to the server or other clients. These

regulations might be made and supervised by either the data holder or even the government (*e.g.* EU General Data Protection Regulation [53] and US Health Insurance Portability and Accountability Act [166]). This arises the need for data decentralization and the data decentralization setup in machine learning is commonly known as federated learning (FL) [115]. FL is a learning paradigm that aims to learn a shared model of interest while keeping the training data stored separately, or in technical terms, *on device*. FL has become a topic of active research in medical image analysis [151, 144, 89, 68, 146, 32]. Unlike the traditional centralized learning environment, the federated setup poses unique challenges for machine learning [154].

Statistical Challenges The goal of FL is to fit a model to a sequence of datasets $\{\mathcal{D}_1, \cdots, \mathcal{D}_m\}$, stored in m distributed clients or data nodes. Each client, $i \in \{1, \cdots, m\}$, collects data in a non-independent and identically distributed (IID) fashion. That is to say, data collected in each client can follow a distinct distribution $\mathcal{D}_i \sim \mathcal{P}_i$. The number of data points on each client may also vary drastically.

System Challenges There are typically a large number of clients, m, in the federated system, and communication is often a non-trivial bottleneck. Moreover, the storage, computational, and communication performance of each client may vary due to hardware capacity. These system challenges, together with statistical heterogeneity, lead to issues such as stragglers [31] and fault tolerance [85] more serious than in data centralization environments.

While efforts have been made to facilitate the research of FL for supervised learning, the impact of data decentralization on PSL remains unclear so far. In fact, there is a lack of a formal definition of the PSL under the federated setup. Thus, in addition to understanding the impact of data decentralization on PSL, this chapter focuses on the following research questions.

- **RQ1**: What is the general problem formulation of federated partially supervised learning?
- **RQ2**: What are the practical concerns of federated partially supervised learning in the medical domain?
- **RQ3**: What are the challenges to solve federated partially supervised learning from a theoretical perspective?

As the partially labeled datasets can be collected from different sources and stored in different locations without direct connections (*e.g.* different hospitals), it is natural to think about the connection between PSL and FL. To answer **RQ1**, I extend the

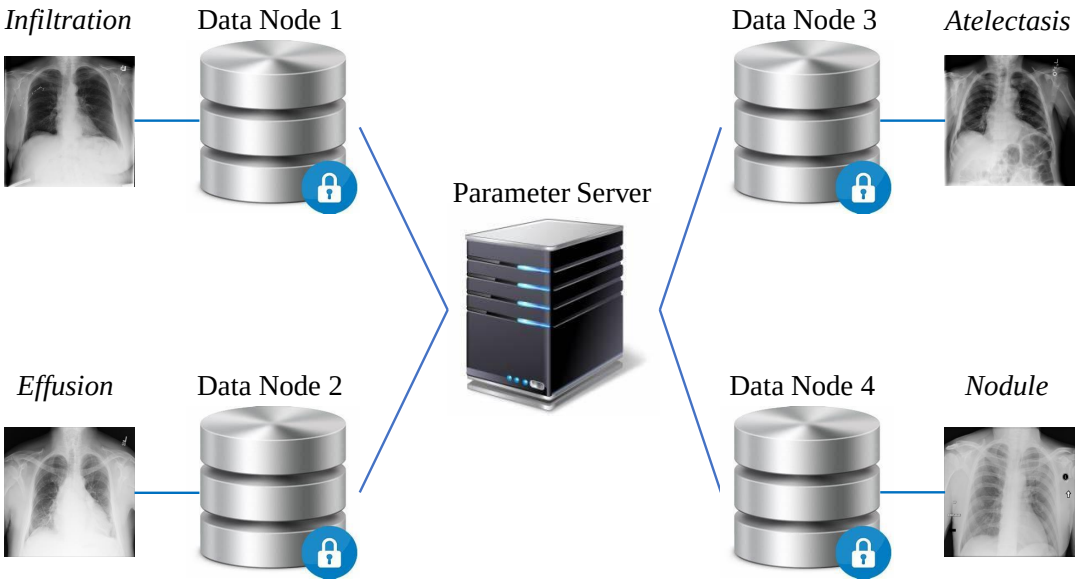


Figure 5.1: Illustration of federated partially supervised learning for a multi-label classification task on chest X-ray images. Here, each client (data node) is annotated for only one thoracic disease. This simplified example is used to convey the main concept of the problem of interest in this chapter. In practice, each client could be partially labeled for more than one class. In the first client, a model can only utilize the label information of *infiltration* for an image but the model does not have access to the label information of the other three diseases assuming that there are only four diseases of interest in total. To ensure data governance, only model weights and the metadata (*e.g.* statistics) of the local data can be communicated between each client and the parameter server (see Section 5.3 for a formal description). In the current scenario, the goal of federated partially supervised learning is to utilize the four partially labeled datasets stored in the four different clients to train the model of interest in the parameter server.

problem formulation of PSL (Section 2.2) 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 Section 5.3. To the best of my knowledge, this is the first study of FPSL. For an intuitive understanding, a concrete example of FPSL is illustrated in Figure 5.1.

Before answering **RQ2**, it is worth noting that a direct combination of FL and PSL does not provide an efficient or robust solution to the problem of interest. Firstly, the federated setup poses a non-trivial barrier for the implementation of certain PSL methods. For example, PSL methods that are based on vicinal risk minimization [39, 45] require access to the training data in a centralized fashion to achieve the

expected efficiency. Another popular family of PSL methods, PSL methods [199] that are based on label propagation involve iterated training over each partially labeled dataset to generate pseudo labels. In contrast to semi-supervised learning, where labels are instance-wise (each instance is either labeled or unlabeled), PSL has task-wise labels (each task is either labeled or unlabeled for a given instance). Thus, in PSL, more iterations should be consumed to ensure the quality of pseudo labels for certain tasks, which leads to a low efficiency in computation under a federated environment. Secondly, in the medical domain, the partially labeled datasets stored in the clients (data nodes) are commonly small, *i.e.* the partial labels are scarce, which means the local data might not be able to support the efficient training of a model with complex network architecture [54, 193]. Thirdly, in FL, the data are assumed to be non-IID. Generally, non-IID describes the situation that clients (*e.g.* hospitals) collect data from different populations. As the data are partially labeled, clients can also have different label distributions. But, none of the existing PSL methods have tried to tackle these challenges directly.

Before presenting the solution, I first examine an existing FL framework, FedAVG [115], which has served as a seminal baseline in FL for fully labeled data. By analyzing FedAVG under FPSL, I conclude that there are two major challenges of FPSL, which answers **RQ3**. Firstly, in the model aggregation step, the aggregation weights should reflect the impact of partial supervision. Secondly, a direct consequence of label scarcity and class imbalance in the local training phase is overfitting. Especially, when each client only has partial labels with respect to a few classes, the features extracted by the learned model might not be able to generalize well to unseen classes. It is important to mitigate local overfitting by learning robust features.

To address the two aforementioned challenges, a simple yet robust FPSL framework `FedPSL` based on FedAVG [115] is presented. `FedPSL` consists of two modules that are designed to address the two challenges, respectively. The first module is a *task-dependent model aggregation* (TDMA) module. The model of interest (a neural network) is decomposed into two parts: a feature extractor and a predictor (*i.e.* a task-specific prediction head). The aggregation of the predictor in the *parameter server* [104] is dependent on the sub-tasks across the clients (*i.e.* each class is considered as a sub-task under the perspective of multi-task learning [17]). The second module is a *task-agnostic decoupling learning* (TADL) module. The local feature extractor could suffer from overfitting caused by both partial supervision and data scarcity. I aim to improve the generalization ability of the local feature extractor by decoupling the learning process of the feature extractor and the predictor. To achieve

this goal, the optimization objective is first reformulated as a *bi-level optimization* problem [4], where the feature extractor and the predictor are optimized on different data splits. Then, by using *meta-learning* [57], I perform meta-optimization on the feature extractor to alleviate the overfitting.

As the first study in FPSL, a primary goal in this work is to provide a comprehensive empirical understanding on FPSL under data scarcity. In addition, I aim to evidence the contributions of `FedPSL`. More specifically, I aim to provide an empirical understanding on the effects of data scarcity and class imbalance under the federated setup, which both can lead to overfitting. Here, the term "data scarcity", also known as "label scarcity" refers to the situation that only limited partial labels are available in the clients. The term "class imbalance" has two meanings: i) the classes with more partial labels can dominate the learning process, and ii) for each class, there are more negative examples than positive examples. Without loss of generality, also for consistency with previous chapters, I illustrate FPSL with multi-label classification (MLC), a representative task prone to overfitting in a federated setup. MLC is a fundamental yet challenging task as it does not have mutually exclusive classes. In contrast to multi-class settings, the constraint of mutually exclusive classes can not be utilized as prior knowledge in either loss formulation [54, 150] or data augmentation [39], which are utilized in centralized PSL. `FedPSL` is evaluated against strong baselines in terms of both performance and robustness under various data challenges such as label scarcity and class imbalance. The empirical results show that `FedPSL` can consistently outperform the baseline methods and can be used as a robust framework for FPSL.

The contributions of this chapter can be summarized as follows:

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

5.2 Related Work

5.2.1 Federated Learning

The concept of FL comes from the necessity of on-device training, where the training data have to remain on the device [115]. FL [124] is contrasted against two related but different fields, distributed learning [171] and edge computing [80] for a better understanding. In this chapter, the definitions of FL, distributed learning, and edge computing follow the standard problem setups [124, 171, 80]. Distributed learning (also known as parallel learning), short for distributed machine learning, denotes the distributed extension of single-node machine learning to handle large-scale training data. In distributed learning, the training data that are centrally available at the beginning are manually distributed to multiple nodes for parallel computing with equal local dataset size and relatively balanced class distribution. The bottleneck of communication in distributed learning comes from the synchronization [31], *i.e.* the trade-off between *synchronous* training and *asynchronous* training. In contrast to synchronous training, where local updates are fully synchronized in a master node, asynchronous training is designed to handle the issue of delayed local updates, or *staleness*. The staleness or straggler issue can be caused by different factors, *e.g.* the computing capacities in the local device and the communication speed between the local device and the parameter server. In this thesis, for simplicity, distributed learning is assumed to have unlimited communication, *i.e.* the parameter server can perform a synchronization when it is required to do so. Meanwhile, edge computing, as suggested by its name, focuses on computing, instead of learning, on the edge device. It is more closely related to the internet of things [7], where sensors collect, process, and analyze various signals. Note, there have been studies that incorporate FL with edge computing [3]. FL features the following characteristics [115].

1. Local optimization: Local training performance plays an important role in global training performance.

2. Non-IID: Any particular user's local dataset will not be representative of the population distribution.

3. Unbalanced: Some clients will have much more local training data than others.

4. Massively distributed: The number of clients participating in optimization is expected to be much larger than the average number of examples per client.

5. Limited communication: The communication might not be stable for the clients all the time.

Following the definitions in [124, 171, 80], a subset of similarities and differences among FL, distributed learning, and edge computing are summarized in Table 5.1 for comparison.

Table 5.1: A subset of similarities and differences among FL, distributed learning, and edge computing.

	Distributed Learning	Edge Computing	Federated Learning
optimization	✓	✗	✓
non-IID data	✗	✓	✓
unbalanced	✗	✓	✓
massively distributed	✗	✓	✓
limited communication	✗	✓	✓

FedAVG [115], proposed by Google, is the first practical FL framework designed for deep learning. Many efforts have been made to improve the performance of FedAVG from different perspectives. FedProx [106] tackles systems heterogeneity and statistical heterogeneity by adding a proximal term to regularize local training. Li *et al*. [107] further theoretically analyze the convergence of FedAVG on non-IID data. FedMA [174] extends the idea of model fusion via optimal transport [153] to federated model aggregation.

However, there are few FL studies directly related to FPSL. Three related areas are federated unsupervised representation learning [202], federated positive-unlabeled learning [187], and federated semi-supervised learning [88]. Following the routine of Chapter 4, I use the terms "federated unsupervised representation learning" and "federated self-supervised learning", interchangeably. FedU [202] present a divergence-aware update mechanism for federated unsupervised representation learning. FedU only considers the local updates rather than global aggregation. As a federated extension of positive-unlabeled learning [11], FedAwS [187] 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 thesis. As semi-supervised learning has been successfully applied to PSL, federated semi-supervised learning is another related domain to FPSL. The state-of-the-art federated semi-supervised learning method FedMatch [88], for instance, adopts a pseudo-labeling training strategy based on consistency regularization. However, FPSL methods that are based on semi-supervised learning exhibit

large variety in terms of class-wise performance, as the quality of pseudo labels are dependent on the amount of available partial labels. This point will be illustrated via experiments in Section 5.6.5.

5.2.2 Partially Supervised Learning under Decentralization

A comprehensive survey of partially supervised learning (PSL), especially PSL on medical images, can be found in Section 2.4. This section mainly focus on the limitations of existing PSL methods under a federated setup. Recently, there have been efforts made to utilize multiple partially labeled datasets in the medical domain. However, none of these methods are designed for the situation that the partially labeled datasets are decentralized. In [39], I address the partial labels issue by generating vicinal labels based on human structure similarity, which can only be implemented in a centralized training environment. [199, 150] both require a fully labeled dataset in the training process. It is infeasible 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 [199] have iterating training procedures, which not only increase the complexity of a federated implementation but also lead to sub-optimal performance. A practical issue that is often ignored in the medical domain is that there are only limited labeled data available. In this work, I denote the situation that only limited partial labels are available in each client as *label scarcity*. PSL methods with complex network architectures or training procedures [54, 193] normally perform much worse than counterparts that have access to large-scale training data in a centralized environment [39]. With small-scale local training data in each client, the issue becomes more challenging in a federated environment. As existing PSL methods struggle in the problem formulation defined in Section 5.3, it is important to study the problem of FPSL and develop a robust solution.

5.2.3 Data Governance

Data governance is a data management concept concerning managing the availability, usability, integrity, and security of the data [92]. In the era of information technology, data privacy has become an emerging topic of active ethical and legal discussion [117]. This chapter specifically focuses on data privacy issues in data governance. Due to the risk of information leakage, data regulations [53, 166] prevent the data holder from transferring the users' data out of local devices in any format, which is the mean

assumption of this chapter. Meanwhile, I want to acknowledge that recent studies have shown that the training data can be recovered from gradients, *e.g.* by gradient inversion [195]. However, the application of privacy-preserving techniques to defend against eavesdropping attacks is beyond the scope of this thesis.

5.3 Problem Setup

I formulate the problem of *federated partially supervised multi-label classification*, an exemplar of FPSL. Assume there are K clients (data nodes) in a federated system and $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$ denotes the partially labeled dataset stored in the k^{th} client[1]. Following standard practice in FL, I assume $\mathcal{D}_k \cap \mathcal{D}_l = \emptyset$ for $k \neq l$ and $\{\mathcal{D}_k\}_{k=1}^K$ are all non-IID data. Given a model of interest f_θ and an independent fully labeled target dataset $\mathcal{S}_{test} = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ that is unseen during the training, the learning outcome is to find the optimal parameter set θ that minimizes the estimated *empirical risk*:

$$\hat{\mathcal{R}_\theta} = \frac{1}{n_t} \sum_{i}^{n_t} \mathcal{L}(f_\theta(x_i^t), y_i^t), \tag{5.1}$$

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

In this thesis, a common FL setup is considered [115], where each data node (client) is only connected to the same master node (server). The master node does not store any data and could be implemented as a parameter server [104]. In addition to the standard setup of FL, the privacy-critical nature of the medical domain imposes another constraint: the transferring of clinical data between the master node and data nodes is prohibited. That is to say, only model weights and *metadata* (*e.g.* statistics of data) [44] should be communicated across nodes. In contrast to the common data privacy issues in FL, the data holders are prevented from exchanging user data in any form to ensure data governance. For example, a hospital might not be allowed to upload its patient data to another institute. With this new constraint, FPSL on medical images is more challenging than a simple integration of FL and PSL.

It is worth mentioning that the primary goal of this work is to formulate FPSL on medical images to facilitate data governance. To obtain privacy-preserving guarantees, an integration of differential privacy [51] techniques such as DPSGD [2] will be required, which, while being out of the scope of this work, is a promising direction for future work.

[1]Note, I only consider the case where are no unlabeled data here for simplicity, following Definition 2.2.4. In addition to the partially labeled datasets, unlabeled datasets $\{\mathcal{U}_k\}_{k=1}^K$ might also be available in each client.

5.4 Towards Understanding Federated Partially Supervised Learning

FPSL is an emerging yet practical problem in the medical domain. Yet, there is limited analysis in the literature. In this section, I provide the first preliminary theoretical understanding of FPSL, which also motivates the proposed method in Section 5.5. I continue to use federated partially supervised multi-label classification as an example.

5.4.1 A Multi-Task Representation of Multi-Label Classification

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

If the multi-label predictor is a fully-connected layer [140], the parameters of the multi-label predictor ϕ can be further represented as a $C \times d$ matrix, where d is the dimension of the output feature vector $f_\theta(x) \in \mathbb{R}^d$ extracted by the feature extractor. In this way, the classification network output can be represented as

$$(g_\phi \circ f_\theta)(x) = \phi \cdot f_\theta(x), \tag{5.2}$$

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

$$\min_{\theta,\phi} \mathcal{L}_\mathcal{S}(g_\phi \circ f_\theta). \tag{5.3}$$

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

$$\phi = \begin{bmatrix} \phi^1 \\ \phi^2 \\ \vdots \\ \phi^C \end{bmatrix}. \tag{5.4}$$

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

$$p_c(x) = \text{sigmoid}(\phi^c \cdot f_\theta(x)), \tag{5.5}$$

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

5.4.2 A Closer Look at FedAVG

Given the formulation of MLC, to understand the difference between FPSL and standard supervised FL, I dissect a seminal FL method, FedAVG [115]. Given $\{\mathcal{D}_k\}_{k=1}^K$ stored in K clients, there are K corresponding multi-label classifiers, denoted as $\{g_{\phi_k} \circ f_{\theta_k}\}_{k=1}^K$. For FedAvg [115], $\{\theta_k\}_{k=1}^K$ and $\{\phi_k\}_{k=1}^K$ are aggregated into θ_0 and ϕ_0 in the parameter server by

$$\begin{aligned} \theta_0 &= \sum_k w_k \theta_k, \\ \phi_0 &= \sum_k w_k \phi_k, \end{aligned} \tag{5.6}$$

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

Without loss of generality, let us look at a single class c. Following the formulation of Equation (5.2) and Equation (5.6), the final prediction model can be written as

$$\begin{aligned} g_{\phi_0^c} \circ f_{\theta_0}(x) &= \phi_0^c \cdot f_{\theta_0}(x) \\ &= \sum_k w_k \phi_k^c \cdot \sum_k w_k f_{\theta_k}(x) \\ &= \sum_{i,j} w_i w_j \phi_i^c \cdot f_{\theta_j}(x). \end{aligned} \tag{5.7}$$

Equation (5.7) is fairly robust under various setups when $\{\theta_k\}_{k=1}^K$ and $\{\phi_k^c\}_{k=1}^K$ are trained with full labels without class imbalance. In a hypothetical scenario, if $\{\theta_k\}_{k=1}^K$ and $\{\phi_k^c\}_{k=1}^K$ both share the same properties of the *oracle* functions θ^* and ϕ^{c*}, Equation (5.7) implies $g_{\phi_0^c} \circ f_{\theta_0}(x) = g_{\phi^{c*}} \circ f_{\theta^*}(x)$.

5.4.3 Challenges of Federated Partially Supervised Learning

The hypothetical scenario in Section 5.4.2, where each client has fully labeled data, is however, rarely the case in practice, especially in the medical domain. Under FPSL, the labels can be partial at each client, which means $\{\theta_k\}_{k=1}^K$ and $\{\phi_k^c\}_{k=1}^K$ might contain noisy information in contrast to standard supervised FL. In addition to partial supervision, each client might not have enough labels for each class to learn meaningful representations.

Again, let us look at a single class c. Assume that $K > 1$ clients are split into two sets, which are $\mathcal{K}_L$ and $\mathcal{K}_U$. $\mathcal{K}_L$ denotes the clients that contain partial labels with respect to class c and $\mathcal{K}_U$ denotes the clients that do not contain any partial labels with respect to class c. A concrete example is the diagnosis of COVID-19 [91]

in China during the early stage of pandemic. While the patients in primary and secondary hospitals might be infected with COVID-19, only limited tertiary hospitals had the capability to diagnose this new disease at that time.

Under this definition, Equation (5.7) can be reformulated as

$$
\begin{aligned}
g_{\phi_0^c} \circ f_{\theta_0}(x) = & \sum_{i_L \in \mathcal{K}_L, j_L \in \mathcal{K}_L} w_{i_L} w_{j_L} \phi_{i_L}^c \cdot f_{\theta_{j_L}}(x) \\
& + \sum_{i_L \in \mathcal{K}_L, j_U \in \mathcal{K}_U} w_{i_L} w_{j_U} \phi_{i_L}^c \cdot f_{\theta_{j_U}}(x) \\
& + \sum_{i_U \in \mathcal{K}_U, j_L \in \mathcal{K}_L} w_{i_U} w_{j_L} \phi_{i_U}^c \cdot f_{\theta_{j_L}}(x) \\
& + \sum_{i_U \in \mathcal{K}_U, j_U \in \mathcal{K}_U} w_{i_U} w_{j_U} \phi_{i_U}^c \cdot f_{\theta_{j_U}}(x).
\end{aligned}
\tag{5.8}
$$

5.4.3.1 Impact on Task-Specific Predictor

Let us assume that f_{θ_k} in Equation (5.8) can achieve the same performance of the *oracle* function and focus instead on the task-specific predictor g_{ϕ_k}. For unlabeled clients, $\mathcal{K}_U$ (with respect to class c), no contributions should be made to the model aggregation as no label information is utilized in the local training. Thus, the third and the fourth terms of Equation (5.8) will inevitably degrade the final performance. At the beginning of the federated training, $\{\phi_k\}_{k \in \mathcal{K}_U}$ are randomly initialized or initialized with unsupervised pre-trained weights. During the local training, $\{\phi_k\}_{k \in \mathcal{K}_U}$ are untouched as no label information is involved in the training of the task-specific predictors. Thus, $\{\phi_k\}_{k \in \mathcal{K}_U}$ are only updated when synchronized with the global model weights, *i.e.* they have to wait to be updated by propagating the error of initialization. Compared with $\{\phi_k\}_{k \in \mathcal{K}_L}$, which are optimized in each local training round, $\{\phi_k\}_{k \in \mathcal{K}_U}$ are de facto *stragglers* [31] in distributed machine learning. This means that $\{\phi_k\}_{k \in \mathcal{K}_U}$ not only lowers the performance but also slows down the convergence.

One might argue that, with large-scale training, FedAVG or advanced variants of FedAVG (*e.g.* FedProx [106]) can automatically mitigate this compound negative effect caused by decentralization and partial supervision. However, it is less practical to collect such large-scale annotations in the medical domain. On the contrary, under the problem formulation of this work, the limited partial labels will further exacerbate the situation.

Moreover, let us consider an extreme case. Say class c is a rare disease and the unlabeled datasets in clients $k \in \mathcal{K}_U$ are relatively larger in size than the partially labeled ones in clients $k \in \mathcal{K}_L$, *i.e.* $w_i \gg w_j \ \forall i \in \mathcal{K}_U, j \in \mathcal{K}_L$. This will make the

fourth term a dominating term in Equation (5.8), which can significantly deteriorate the FL system, for the same reasons mentioned above.

5.4.3.2 Impact on Task-Agnostic Feature Extractor

Now, let us relax the assumption in Section 5.4.3.1 and analyze the behavior of f_θ. In standard supervised learning, θ and ϕ are optimized together by minimizing Equation (5.1). Thus, under a *Bayesian* view, the prediction can also be represented as $(g_{\phi^c|\theta} \circ f_\theta)(x)$, *i.e.* ϕ^c is dependent on θ (or in another direction, θ is dependent on ϕ^c). With large-scale fully labeled data, such dependence is normally ignored, as in each client, f_θ should extract information for all classes of interest more or less. However, under FPSL, the clients $\mathcal{K}_U$ are not optimized to capture features with respect to c. This can cause non-trivial *weight divergence* [197] across the clients $\mathcal{K}_U$ and the clients $\mathcal{K}_L$ in the system.

(5.6) can also be interpreted as *ensemble learning* [35]. Ensemble learning aims to improve the model generalization ability by leveraging multiple classifiers [200]. However, under FPSL, due to partial supervision, the quality of f_θ in a client could be biased to a few classes. While this may have a positive effect on the first term of Equation (5.8) for class c, the other classes can be negatively influenced by the last three terms. Again, with large-scale labeled data, this issue can be mitigated. However, with limited data, the overfitting could be severe. Ideally, θ should be task-agnostic to improve the generalization ability of the feature extractor.

5.5 Method

Motivated by the limitations of FedAVG in Section 5.4.3, I propose `FedPSL`, an FPSL framework consisting of two simple yet efficient techniques that aim to better handle the weight assignment and efficiently utilize the limited partial supervision. First, a model aggregation module is proposed in Section 5.5.1 by considering the number of partial labels. Second, I propose to decouple the feature extractor and predictor in the learning process in Section 5.5.2. The empirical advantages of `FedPSL` will be illustrated in Section 5.6.

5.5.1 Task-Dependent Model Aggregation

As discussed in Section 5.4.3.1, the model aggregation on the predictor should be linked with class-wise partial labels across the clients, instead of the amount of data. To mitigate the negative impact of *wrong* weight assignments in FedAVG, the model

aggregation is conducted separately for the feature extractor and the multi-label predictor. For the feature extractor f_θ, I adopt the same aggregation mechanism as Equation (5.6). For the multi-label predictor g_ϕ, the task-specific weights ϕ^c are aggregated independently:

$$\begin{aligned} \theta_0 &= \sum_k \frac{n_k}{\sum_k n_k} \theta_k, \\ \phi_0^c &= \sum_k \frac{n_k^c}{\sum_k n_k^c} \phi_k^c, \end{aligned} \tag{5.9}$$

where n_k^c denotes the number of labeled examples in client k with respect to class c. Note, n_k in Equation (5.6) and n_k^c in Equation (5.9) are all metadata. Exchanging metadata between clients or uploading metadata to the parameter server is also known as *metadata transfer* [44].

In contrast to Equation (5.6), which is designed for standard supervised learning, I use the number of partially labeled examples with respect to class c in each client to indicate the contribution of this client. For clients that do not contain label information of c, the contributions to the model aggregation will just be zeros. However, I still use n_k, instead of $\sum_c n_k^c$, to indicate the contribution to θ. This is because $\sum_c n_k^c \geq n_k$, which could exacerbate the negative effect of class imbalance on the feature extractor and the proposed solution aims to alleviate it. For the same reason, the feature extractor and the multi-label predictor should be decoupled in the local training to improve the generalization ability.

5.5.2 Task-Agnostic Decoupling Learning

As discussed in Section 5.4.3.2, the feature extractor f_θ and the multi-label predictor g_ϕ are mutually dependent. This means that the training of f_θ could get biased to the partially labeled classes in each client. To mitigate the local overfitting and improve the generalization ability of f_θ, I propose a local training strategy based on *meta-learning* [57].

Before presenting the training strategy, I first rephrase the local training target as a *bi-level optimization* [4, 26] problem. Assume the local training data $\mathcal{D}_k$ can be split into two subsets, one is denoted as the training set and the other one is denoted as the validation set. I propose the following formulation:

$$\begin{aligned} &\min_\theta \mathcal{L}_{val}(g_{\phi^*|\theta} \circ f_\theta(x), y); \\ &\text{s.t.} \qquad \phi^*|\theta = \arg\min_\phi \ \mathcal{L}_{train}(g_\phi \circ f_\theta(x), y), \end{aligned} \tag{5.10}$$

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

It is worth mentioning that the formulation Equation (5.10) only makes sense under data scarcity. Given large-scale fully labeled data, Equation (5.10) actually leads to less inefficient training than standard joint supervised training, as θ and ϕ are optimized in different feature spaces. However, with only limited partially labeled data, Equation (5.10) can be linked with meta-learning to improve the generalization ability of f_θ. The theoretical advantages of using meta-learning to learn shared representations under MTL have been actively discussed in [57, 128, 78].

A direct implementation of Equation (5.10) is difficult and suffers from inefficient training. Thus, I propose to approximate Equation (5.10) with *stochastic gradient descent*. The idea is to approximate ϕ^* by adapting θ using only a single training step, without solving the inner optimization completely. The local training scheme in each client is illustrated in Algorithm 5.1. Note, in Line $6-7$, the gradients are evaluated with respect to both θ_k and ϕ_k with $\mathcal{B}_{train} \in \mathcal{D}_k$, instead of ϕ_k alone in Equation (5.10). Line 9 depicts the meta-optimization step, where the meta-objective is

$$\begin{aligned} &\min_\theta \mathcal{L}_{\mathcal{B}_{val}}(g_{\phi_k} \circ f_{\theta_k'}) = \\ &\qquad \min_\theta \mathcal{L}_{\mathcal{B}_{val}}(g_{\phi_k} \circ f_{\theta_k - \alpha \nabla_{\theta_k} \mathcal{L}_{\mathcal{B}_{train}}(g_{\phi_k} \circ f_{\theta_k})}) \end{aligned} \tag{5.11}$$

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

The complete federated training scheme is presented in Algorithm 5.2.

5.6 Experiments

The purposes of the conducted experiments are threefold. Firstly, I aim to illustrate that FPSL is an under-explored yet challenging problem compared with standard FL and centralized training. Secondly, I want to discuss several initial solutions to FPSL. Thirdly, I want to demonstrate the robustness of `FedPSL` against label scarcity and class imbalance. I use a multi-label classification (MLC) task on chest X-ray images to evaluate the proposed framework. The labels for an MLC task are usually sparse (*e.g.* 60% of chest X-ray images in ChestX-ray14 dataset [177] have no findings of thoracic diseases), which makes federated partially supervised multi-label classification even more difficult in practice.

Algorithm 5.1 Local training procedure for client k.

$\mathcal{S}_k$: Partially labeled data in client k
E: Number of epochs
B: Number of batches
α, β: Learning rates

function CLIENT_UPDATE(θ_k, ϕ_k)
  for $t = 1, 2, \cdots, E$ **do**
    for $b = 1, 2, \cdots, B$ **do**
      Sample $\mathcal{B}_{train}, \mathcal{B}_{val}$ from $\mathcal{D}_k$
      // Update ϕ_k with $\mathcal{B}_{train}$
      $\theta_k^{'} \leftarrow \theta_k - \alpha \nabla_{\theta_k} \mathcal{L}_{\mathcal{B}_{train}}(g_{\phi_k} \circ f_{\theta_k})$
      $\phi_k \leftarrow \phi_k - \alpha \nabla_{\phi_k} \mathcal{L}_{\mathcal{B}_{train}}(g_{\phi_k} \circ f_{\theta_k})$
      // Meta-update θ_k with $\mathcal{B}_{val}$
      $\theta_k \leftarrow \theta_k - \beta \nabla_{\theta_k} \mathcal{L}_{\mathcal{B}_{val}}(g_{\phi} \circ f_{\theta_k^{'}})$
  return θ_k, ϕ_k

Algorithm 5.2 Training procedure for `FedPSL`. "PS" denotes the parameter server. "Client_Update" is a function that performs local partially supervised training.

M: Metadata
T: Number of training rounds

for $t = 1, 2, \cdots, T$ **do**
  for $k = 1, 2, \cdots, K$ **do**
    $\theta_k, \phi_k \leftarrow \theta_0, \phi_0$ ▷ Synchronize with PS
    $\theta_k, \phi_k \leftarrow$ Client_Update(θ_k, ϕ_k)
  Upload $\{\theta_k\}_{k=1}^K$ and $\{\phi_k\}_{k=1}^K$ to PS
  $\theta_0, \phi_0 \leftarrow$ Aggregate$(\{\theta_k\}_{k=1}^K, \{\phi_k\}_{k=1}^K, M)$ ▷ Equation (5.9)

5.6.1 Datasets

Three public datasets are utilized to simulate the multi-site partially labeled datasets. The visual difference on the multi-site chest X-ray images is illustrated in Figure 5.2.

5.6.1.1 ChestX-ray14 Dataset

ChestX-ray14 dataset [177] is a public chest X-ray image dataset for multi-label chest disease detection. The dataset contains 112120 chest X-ray images and contains label information of 14 classes. Each chest X-ray image can contain multiple diseases at the same time. Each image has a fixed resolution of 1024×1024. The dataset can be downloaded from `https://nihcc.app.box.com/v/ChestXray-NIHCC/`.



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

Figure 5.2: Visual illustration of multi-site chest X-ray images. (a) is diagnosed with cardiomegaly and emphysema. (b) is tuberculosis positive. (c) is COVID-19 positive.

5.6.1.2 Tuberculosis Chest X-ray Database

Tuberculosis Chest X-ray Database (Tuberculosis dataset) [138] is a public chest X-ray image dataset, where each chest X-ray image is only annotated for tuberculosis. The dataset contains 700 tuberculosis and 3500 normal chest X-ray images. The tuberculosis and normal chest X-ray images are collected and curated from multiple sources and have various resolutions. The dataset can be downloaded from `https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-dataset`.

5.6.1.3 COVID-19 Detection Dataset

COVID-19 Detection Dataset (COVID-19 dataset) [25] is a public chest X-ray image dataset, where each chest X-ray image is only annotated for COVID-19. The dataset contains 3616 COVID-19 and 10192 normal chest X-ray images. The COVID-19 and normal chest X-ray images are collected and curated from multiple datasets. Each image has a fixed resolution of 299×299. The dataset can be downloaded from `https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database`.

5.6.2 Experimental Setup

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

In this chapter, I choose the AUROC (see Section 2.3.2.2 for details of AUROC) as the evaluation metric as the overall federated learning performance is important

(see Section 2.3.2.3 for the reasons). I select the best performance in each run based on the highest average AUROC of the C diseases. The mean over three runs with different random seeds is reported.

5.6.3 Baselines

The choice of baseline methods gives consideration to two aspects. The first aspect is to provide an empirical understanding of FPSL and the second aspect is to examine the performance of the seminal methods from existing learning paradigms such as supervised learning, semi-supervised learning, and self-supervised learning, when addressing FPSL.

`FedPSL` is compared against six robust FPSL baselines.

- `FedAVG` is an adaptation of FedAVG [115] to FPSL. To differentiate from FedAVG, I use `FedAVG` to denote the method. FedAVG is a seminal method that has robust performance in FL. In the clients, only the weights with respect to the given partial labels in the backpropagation are updated. Note, this is equivalent to standard supervised learning. In the parameter server, the shared weights are aggregated and synchronized. As the multi-label predictor g_ϕ is implemented as a fully-connected layer in DenseNet121, all model weights are shared.

- `FedProx` is an adaptation of FedProx [106] to FPSL. To differentiate from FedProx, which is designed for non-IID fully labeled data, I use `FedProx` to denote the method. Following the same setup of `FedAVG`, I use 0.001 for the proximal term. I use the official implementation from `https://github.com/litian96/FedProx`.

- `FedSSP` denotes a learning paradigm of self-supervised pre-training followed by fine-tuning on partial labels. A state-of-the-art self-supervised learning framework, SimSiam [23], is adapted to a federated environment. In the pre-training phase, only the feature extractor f_θ is pre-trained. Then, the feature extractor and the prediction heads are fine-tuned based on partial labels, in a fashion similar to `FedAVG`. I use the official implementation from `https://github.com/facebookresearch/simsiam` and the default hyperparameters are adopted.

- `FedMatch` is an adaptation of FedMatch [88], a state-of-the-art federated semi-supervised learning baseline based on consistency regularization training. As

there is no existing PSL method designed for the problem of interest yet. I use FedMatch as a strong PSL baseline. I use the official implementation from `https://github.com/wyjeong/FedMatch` and the default hyperparameters are adopted.

- `FedMatch+` follows the same setup of `FedMatch`, except that the feature extractor is pre-trained in the same way as `FedSSP`.

The second group includes four centralized baselines, where the corresponding constraints in Section 5.3 are relaxed (*i.e.* centralized training is feasible or full labels are available). Three centralized PSL methods and a supervised *Oracle* with full labelsare included to provide an empirical understanding of the impact of FPSL .

- IML [63] is the simplest PSL method, which simply ignores missing labels, *i.e.* only backpropagating the gradients corresponding to the partial labels.

- FixMatch is a centralized semi-supervised learning method, which is adapted to PSL for the same reason of `FedMatch`. The same set of hyperparameters of `FedMatch` is used. This is also the centralized training counterpart of `FedMatch`. I use the official implementation from `https://github.com/google-research/fixmatch` and the default hyperparameters are adopted.

- MixUp-PME [45] is a centralized PSL method based on data augmentation and pseudo labeling. The default hyperparameters of [45] are adopted.

- *Oracle* is the standard supervised learning baseline, which has centralized data with full labels. As expected, this is the best performance that the model of interest can achieve under standard supervised learning, which is also considered as the upper bound performance for centralized fully labeled data.

5.6.4 Implementation

5.6.4.1 Data Pre-Processing

Each chest X-ray image has a resolution of 1024×1024. During both training and testing of the model, each chest X-ray image is first resized to 224×224 and the image is then normalized by instance normalization [30] in Equation (3.6). The data augmentation policy proposed in Listing 4.1 is used for pre-training.

5.6.4.2 Network Architecture

Following [140], all baseline methods use a DenseNet121 [81] as the encoder f_θ. In addition, DenseNet121 is a commonly adopted model in FL [115] for a lightweight experimental setup. The experiments with more advanced architectures are considered out of the scope of this work. Each of the federated methods has $K+1$ DenseNet121s for K clients and the parameter server, while each of the non-federated methods has one DenseNet121. All models are implemented in PyTorch [132] on an NVIDIA Tesla V100.

5.6.4.3 Training

For a fair comparison, all networks are initialized with the same random seeds. All methods are trained with partial or full labels for 100 epochs. The synchronization and aggregation for the federated methods are performed every 10 epochs. A standard Adam [97] optimizer with a fixed learning rate 10^{-3} is used for supervised training or partially supervised training. For `FedPSL`, I set $\alpha = \beta = 10^{-3}$, *i.e.* the same learning rates are used for the standard supervised learning setup and the meta-learning setup. I choose these hyperparameters following the common practice in large-scale optimization [74]. The binary cross-entropy loss in Equation (3.3) for each class is weighted by $\frac{N_{neg}}{N_{pos}+N_{neg}}$ and $\frac{N_{pos}}{N_{pos}+N_{neg}}$, where N_{neg} and N_{pos} are the numbers of negative cases and positive cases for the class of interest in the labeled data. The PyTorch bi-level optimization package `higher` is used to perform the meta-optimization step. I use the official implementation from `https://github.com/facebookresearch/higher`.

5.6.5 Federated System without One-Class Clients

5.6.5.1 Data Setup

In the first experiment, a common scenario is considered, where each client has $1 < C_k < C$ classes, *i.e.* each client has more than one class. Furthermore, I assume that the domain shifts across the clients are negligible. Here, I use $K = 4$ clients and $C = 6$ classes to simulate the situation. Four batches (batch 2 - batch 5, with default batch splits) of ChestX-ray14 dataset are used to create four clients. Each client contains 10^3 partially labeled chest X-ray image images. The label statistics are presented in Table 5.2. I also prepare 300 positive cases and 300 negative cases for each of six classes from other batches as an independent balanced test set.

Table 5.2: Label statistics of positive cases for the experimental setup **without** one-class clients in the federated system. "0" denotes that the client does not have any partial labels for the disease, *i.e.* this disease is unlabeled.

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

5.6.5.2 Impact of Partial Supervision

The numerical results are presented in Table 5.3. Compared with the *Oracle*, the fully supervised baseline, all PSL methods (both centralized and decentralized) are negatively impacted, as PSL methods only receive partial labels. However, PSL methods can achieve competitive performance on certain diseases, even with partial labels, *e.g.* "Consolidation" and "Pneumothorax". This means that PSL methods do make a difference where there are limited training examples as they can extract meaningful information even without full supervision (with respect to certain classes).

5.6.5.3 Analysis of Centralized PSL Methods

Recall that IML [63] is the centralized baseline that does not utilize any advanced techniques (*e.g.* pre-training, pseudo labeling, regularization). As shown in Table 5.3, MixUp-PME [45] improves the overall performance of IML, while it slightly decreases the performance of several diseases compared to IML. I conjecture that the performance of MixUp-PME is class-dependent and can further be improved with sophisticated training tricks such as the ones described in [45]. FixMatch [157] performs even worse than IML as a state-of-the-art pseudo-label based semi-supervised learning method. A similar phenomenon is also reported in [39], where pseudo labeling-based PSL methods are less robust with only limited labeled data, *i.e.* they fail to generate high-quality pseudo labels.

5.6.5.4 Impact of Decentralization

Based on Table 5.3, decentralization has a positive impact on PSL: when comparing the centralized and decentralized methods, decentralized PSL methods outperform centralized PSL methods by a large margin. This is actually counter-intuitive, as

under standard supervised learning, decentralization usually slows the convergence rate and lowers the overall performance. As all methods are trained by stochastic gradient descent with random mini-batch sampling, I conjecture that centralization can amplify the negative impact of partial supervision. In a centralized environment, a random mini-batch could contain partially labeled examples from several partially labeled datasets. The partial supervision can cause inefficient training during back-propagation, especially when C is large. In contrast, decentralization slightly alleviates this issue. In each client, each mini-batch only contains relevant partial labels. This process can be more efficiently than centralized training for the model to extract transferable representations for only a few classes. Besides, in a centralized environment, when there are "dominating" classes (classes with more partial labels) or "easy" classes (classes with lower learning difficulty), the learning process will be inevitably influenced by such class imbalance. Meanwhile, decentralization ensures that local training is conducted independently and class imbalance is somewhat mitigated by model aggregation.

5.6.5.5 Comparison with Federated Baselines

`FedAVG` and `FedMatch+` are the two best performing FPSL baselines considering average F1-score. While *self*-SL has played an important role in label-efficient learning, self-supervised pre-training shows different influences on `FedAVG` and `FedMatch`, a state-of-the-art federated semi-supervised learning method. The federated pre-training slightly improves the performance of `FedAVG` on a few classes at the cost of decreased performance on other diseases. On the contrary, `FedMatch` significantly benefits from federated pre-training on all classes. As mentioned before, semi-supervised learning suffers from label scarcity to produce high-quality pseudo labels. The results show that `FedMatch+` can leverage pre-training to deal with label scarcity. It is worth mentioning that `FedMatch+` (*i.e. self*-SL and semi-supervised learning) does significantly improve the performance on several classes over the `FedAVG` baseline. However, similar to `FedSSP`, the performance of a few classes are negatively influenced. In contrast to standard image recognition tasks, such as CIFAR-10 and CIFAR-100 [100], which are commonly used in FL studies and assumed to have similar learning difficulties, medical images require additional consideration. In addition to the number of partial labels for each class, I hypothesize that the performance of `FedMatch+` is also determined by the learning difficulty of each class, which is not reflected in Section 5.4.3.

5.6.5.6 Analysis of FedPSL

FedPSL, while being based on FedAVG, outperforms all FPSL baselines by a large margin. Meanwhile, notice that FedPSL can even outperform *Oracle* on "Consolidation". To have a better understanding on the credit assignment of the two proposed modules, I further include ablation models that either include only the task-dependent model aggregation (TDMA) module or only the task-agnostic decoupling learning (TADL) module. These ablation models are denoted as FedPSL *w/ TDMA* and FedPSL *w/ TADL*, respectively. The two modules both improve the performance of FedAVG, with TADL having a larger impact. However, as shown in Algorithm 5.1, TDMA requires less computation to achieve better performance than TADL and is thus more computationally efficient.

5.6.6 Federated System with One-Class Clients

5.6.6.1 Data Setup

In the second experiment, a different scenario is considered, where each client has $1 \leq C_k < C$ classes, *i.e.* there can be one-class clients. In contrast to multi-class clients, the one-class client commonly collects data for a specific purpose, *e.g.* a less common disease which is unseen in other clients. For these less common diseases, the imaging protocols of these chest X-ray images could be different from the common diseases, which leads to domain shifts (see Figure 5.2). Thus, I further assume that domain shifts exist across the clients. Here, I use $K = 6$ clients and $C = 8$ classes to simulate the situation. I use the four batches (batch 2 - batch 5, with default batch splits) of the ChestX-ray14 dataset to create the first four clients, and use the Tuberculosis and COVID-19 datasets to create two one-class clients. Each client contains 10^3 partially labeled chest X-ray image images. The label statistics are presented in Table 5.4. I also prepare 300 positive cases and 300 negative cases for each of eight classes from other batches as an independent balanced test set.

5.6.6.2 Impact of One-Class Clients

Based on the empirical results in Table 5.3, two best-performing federated PSL methods FedAVG and FedMatch+ are chosen as the baselines. The numerical results are presented in Table 5.5. When comparing the results for this experiment (Table 5.5) with Table 5.3, I observe an interesting phenomenon: with new one-class clients added into the federated system, the *Oracle* tends to get lower performance on the six diseases in Table 5.3. At first glance, this appears counter-intuitive, as usually, more

Table 5.3: Quantitative evaluation of multi-label thoracic disease classification in a federated system **without** one-class clients. The reported number are the mean and standard deviation of average AUROCs over three random seeds. Blue denotes the highest number in each column (other than the *Oracle*). The column "**Average**" denotes the average AUROCs over classes, which is considered as the overall performance for a particular method.

	Method	Infiltration	Effusion	Atelectasis	Nodule	Consolidation	Pneumothorax	**Average**
Decentralized	`FedAVG`	0.648 ± 0.027	0.704 ± 0.015	0.690 ± 0.007	0.731 ± 0.021	0.814 ± 0.012	0.800 ± 0.015	0.731 ± 0.008
	`FedProx`	0.604 ± 0.049	0.719 ± 0.003	0.673 ± 0.010	0.762 ± 0.037	0.819 ± 0.011	0.793 ± 0.014	0.728 ± 0.009
	`FedSSP`	0.618 ± 0.065	0.734 ± 0.015	0.697 ± 0.005	0.632 ± 0.008	0.799 ± 0.005	0.702 ± 0.014	0.697 ± 0.011
	`FedMatch`	0.676 ± 0.048	0.710 ± 0.044	0.717 ± 0.042	0.562 ± 0.016	0.761 ± 0.041	0.656 ± 0.030	0.680 ± 0.032
	`FedMatch+`	0.693 ± 0.008	0.778 ± 0.008	0.748 ± 0.009	0.610 ± 0.012	0.824 ± 0.007	0.761 ± 0.002	0.736 ± 0.003
Centralized	IML [63]	0.599 ± 0.064	0.711 ± 0.009	0.662 ± 0.026	0.570 ± 0.022	0.717 ± 0.015	0.614 ± 0.046	0.646 ± 0.010
	FixMatch [157]	0.600 ± 0.017	0.700 ± 0.016	0.637 ± 0.014	0.515 ± 0.013	0.756 ± 0.004	0.559 ± 0.025	0.628 ± 0.005
	MixUp-PME [45]	0.561 ± 0.029	0.778 ± 0.016	0.721 ± 0.020	0.586 ± 0.045	0.713 ± 0.040	0.596 ± 0.040	0.659 ± 0.013
Ours	`FedPSL` *w/ TDMA*	0.662 ± 0.017	0.757 ± 0.007	0.662 ± 0.064	0.719 ± 0.093	0.845 ± 0.009	0.799 ± 0.030	0.741 ± 0.009
	`FedPSL` *w/ TADL*	0.692 ± 0.018	0.765 ± 0.007	0.720 ± 0.067	0.731 ± 0.091	0.848 ± 0.012	0.807 ± 0.015	0.757 ± 0.009
	`FedPSL`	0.696 ± 0.017	0.771 ± 0.007	0.720 ± 0.059	0.738 ± 0.081	0.848 ± 0.011	0.803 ± 0.015	0.763 ± 0.011
Oracle		0.868 ± 0.010	0.907 ± 0.025	0.931 ± 0.015	0.792 ± 0.002	0.823 ± 0.009	0.809 ± 0.004	0.855 ± 0.010

Table 5.4: Label statistics of positive cases for the experimental setup **with** one-class clients in the federated system. "0" denotes that the client does not have any partial labels for the disease, *i.e.* this disease is unlabeled.

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

labeled data should lead to better performance. Meanwhile, federated PSL methods all achieve robust overall performance ("**Average**") and can even outperform the *Oracle* on several classes (*e.g.* "Nodule" and "Tuberculosis"). A good example is "Tuberculosis". The chest X-ray images of the Tuberculosis dataset are visually different from the chest X-ray images of the Chest-Xray14 dataset, as shown in Figure 5.2. As the classes of the one-class clients are unseen in the other clients, the *Oracle* and PSL methods in fact have the same training data for 'Tuberculosis". However, the *Oracle* performs worse. This suggests that the feature extractor of the *Oracle* is negatively influenced by this newly added disease with obvious domain shift. This indicates that federated PSL methods are robust against the one-class clients and can even benefit from certain level of decentralization.

5.6.6.3 Analysis of `FedPSL`

As in the previous experiment (Table 5.3), `FedPSL` achieves the best overall performance and competitive performance on every single class. Notably, `FedPSL` outperforms all other methods, including *Oracle*, on "Tuberculosis" and "COVID-19", by a large margin. As these two classes are only present in two different clients, this further validates the discussion in Section 5.4.3 regarding the necessity of TDMA. Compared with the *Oracle*, the results further suggest that severe overfitting is a main challenge under data scarcity, where TADL is a more efficient solution than TDMA.

Table 5.5: Quantitative evaluation of multi-label thoracic disease classification a federated system **with** one-class clients. The reported number are the mean and standard deviation of average AUROCs over three random seeds. Blue denotes the highest number in each column (other than the *Oracle*). The column "**Average**" denotes the average AUROCs over classes, which is considered as the overall performance for a particular method.

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

5.6.7 Robustness Analysis

In addition to Section 5.6.5 and Section 5.6.6, a few ablation experiments are further designed to evaluate `FedPSL` under various situations, namely under different levels of data scarcity, with extreme clients, and under different levels of synchronization frequency. For simplicity, I use the data setup in Section 5.6.5.1, where the federated system has $K = 4$ clients and $C = 6$, *i.e.* without one-class clients. `FedAVG` is chosen as the baseline in this section for its robust performance in previous experiments.

5.6.7.1 Sensitivity to Data Scarcity

In the first ablation experiment, I study the impact of data scarcity to `FedPSL`. Here, data scarcity describes the situation that each client only has access to small amounts of data. I denote that each client has n partially labeled examples (I sample the first n examples in each batch). The relation between the overall performance and n is depicted in Figure 5.3. `FedPSL` shows obvious advantages over `FedAVG` when n is small, while the performance gain diminishes as n increases. This matches the expectations given the discussion in Section 5.5: TDMA and TADL are designed for data scarcity situations. As discussed in Section 5.5.2, with large-scale data, using TADL could lead to inefficient training. A potential solution is hybrid training (standard training for clients with large local datasets and TADL for clients with small local datasets), an exploration of which is considered beyond the scope of this work.

5.6.7.2 Sensitivity to Extreme Clients

In the second ablation experiment, the impact of extreme clients to `FedPSL` is studied. In the previous experimental setups, I tend to assume that each client has access to the same number of samples. Here, instead, the *extreme* clients denote the ones that have significantly smaller or larger local datasets. While following the previous setup, I choose the second and the third clients to be the extreme clients. As each batch of the Chest-Xray14 dataset has 10^4 chest X-ray images, each client can have up to 10^4 chest X-ray images. Again, I use n to denote the size of the local datasets in the extreme clients. The relation between the overall performance and n is depicted in Figure 5.4. `FedPSL` can achieve more robust performance that `FedAVG`. By comparing Figure 5.3 and Figure 5.4, I can also infer that, when the federated systems have roughly the same sizes of total data, the system with extreme clients has lower performance than the system without extreme clients.

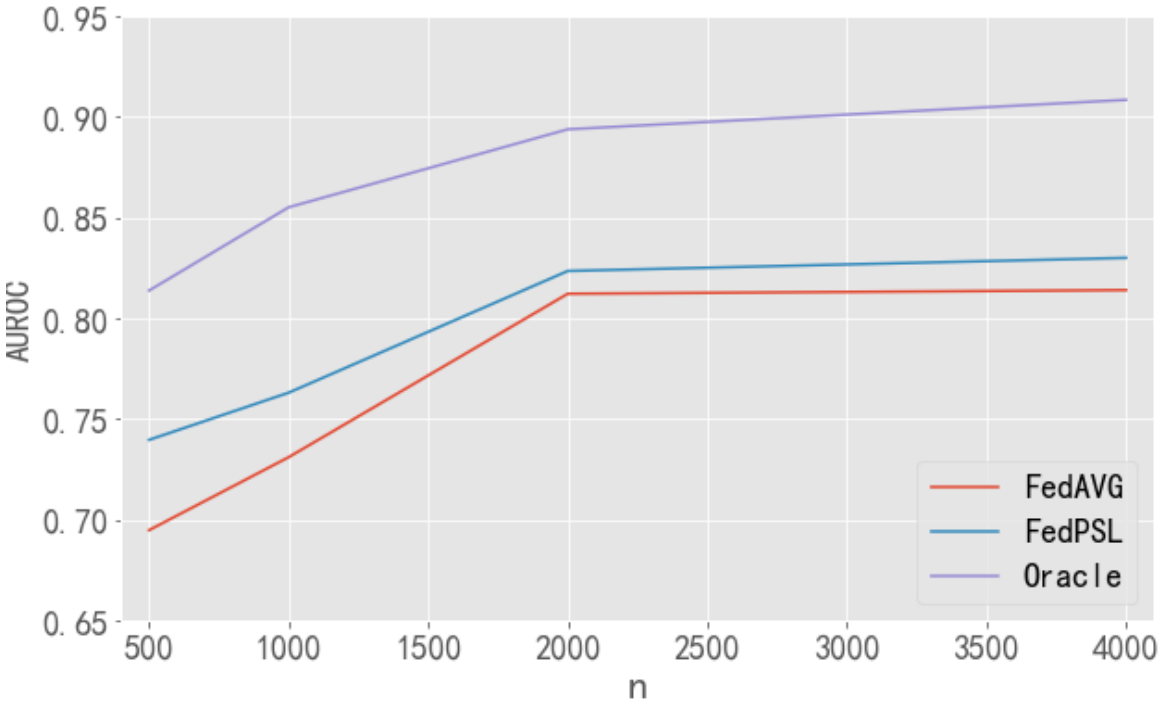


Figure 5.3: Sensitivity of the overall performance (AUROC) to the size of the local dataset (n). The overall performance is measured by the mean AUROC over $C = 6$ classes in $K = 4$ clients.

5.6.7.3 Impact of Synchronization Frequency

An important hyperparameter in FL, which is often ignored, is the synchronization frequency. In practice, it is unlikely to exchange model parameters instantly (*e.g.* directly exchanging gradients in distributed machine learning [31]). I perform an ablation study on the synchronization frequency. I use E to denote the number of local training epochs before synchronization. The results are summarized in Table 5.6. In comparison with `FedAVG`, `FedPSL` is more robust against low synchronization frequency, while the performance gap diminishes as E grows up. However, `FedPSL` exhibits a higher standard deviation than `FedAVG`.

5.7 Conclusion

In this chapter, I formulate and discuss a new research problem, federated partially supervised learning for limited decentralized partially labeled medical images. I theoretically explain the challenges of federated partially supervised learning under data scarcity. To address the challenges, I propose `FedPSL`, a simple yet robust solution to federated partially supervised learning. `FedPSL` consists of two novel modules, a

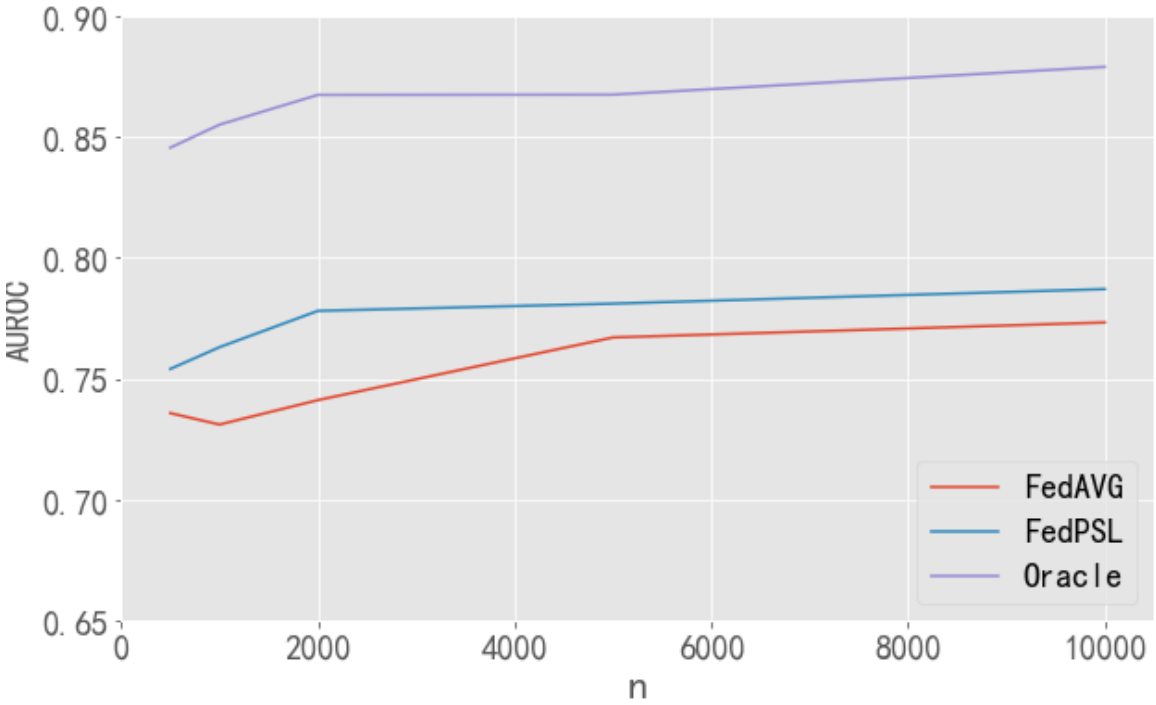


Figure 5.4: Impact of extreme clients on the overall performance (AUROC). n denotes the size of the local dataset in the extreme clients, while each common client has 1000 partially labeled examples. Two out of four clients are the extreme ones. The overall performance is measured by the mean AUROC over $C = 6$ classes in $K = 4$ clients.

task-dependent model aggregation module to address the aggregation weight assignment issue and a task-agnostic decoupling learning module based on meta-learning to address the local overfitting issue. Finally, I provide an empirical understanding of federated partially supervised learning for the first time. The experimental results also indicate a new research direction in label-efficient learning with partial supervision.

Table 5.6: Impact of synchronization frequency on the overall performance (AUROC). The reported numbers are the mean and standard deviation of average AUROCs over three random seeds. The overall performance is measured by the mean AUROC over $C = 6$ classes. E denotes the number of local training epochs. Blue denotes the highest number in each column (other than the *Oracle*).

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

Chapter 6

Federated Partially Supervised Learning for Rare Classes

A long-standing challenge in machine learning is class imbalance. Given the same population, different diseases can have different prevalence and incidence, *i.e.* the numbers of positive cases for the diseases of interest can be drastically different. This chapter discusses an instance of the extreme class imbalance issue in federated partially supervised learning. In federated partially supervised learning, each client can have annotations of different classes of interest. This chapter focuses on *rare* diseases, which are less common than most of diseases in the population. Due to the rareness, only a few clients may have access to the annotations of these rare diseases. Moreover, each of these clients may have only few labeled examples, *i.e.* rare classes have many fewer cases than *common* classes. A concrete example of a federated system in the medical domain is that general hospitals are more familiar with common diseases and only specific research institutes have knowledge of rare diseases. However, existing federated learning methods for partially supervised multi-label classification of common classes and rare classes are inefficient. Due to extreme class imbalance and decentralization, improving the performance of rare classes can inevitably decrease the learning performance of common classes, while maintaining the performance of common classes might lead to rare classes being completely ignored. Thus, the problem of interest in this chapter can be rephrased as how to leverage common classes to learn rare classes without impairing the performance of common classes in a federated system. This chapter not only formulates the problem of interest for the first time but also offers the first solution to it.

6.1 Introduction

In Chapter 5, I formulate the problem of federated partially supervised learning for the first time and introduce the challenges of federated partially supervised learning, especially from the perspective of data decentralization and federated optimization. In federated partially supervised learning, partially labeled datasets are decentralized and stored separately in different clients. In comparison with fully labeled datasets in traditional federated learning, clients can now have different classes labeled.

Meanwhile, one critical issue unsolved in partially supervised learning (PSL) is class imbalance. In Section 2.2.2.2, the definitions of *intra-class* class imbalance and *inter-class* class imbalance have been discussed. In this chapter, the focus is the extreme *inter-client* class imbalance, which is a federated extension of *inter-class* class imbalance. In a *open-world* scenario, there are newly-found or under-examined classes, which tend to have much fewer labeled instances than well-understood classes. In this chapter, the former classes are denoted as *rare* classes or *underrepresented* classes, and the latter classes are denoted as *common* classes. In this chapter, the terms "rare classes" and "underrepresented classes" are used interchangeably. While a formal problem setup will be described in Section 6.3, an intuitive illustration of the problem is presented in Figure 6.1.

Given two types of classes defined above that directly describe the extreme class imbalance between classes, this chapter discusses an unexplored setup of federated partially supervised learning. It is practical to have the following assumption that rare classes show up in fewer clients than common classes and rare classes have *much* fewer labeled examples in each client than common classes. Note, common classes and rare classes are annotated at disjoint clients, which makes this practical problem even more challenging. Considering a concrete example, when a new disease is found, only specific clinical research institutes have the ability to collect the patients' data and diagnose this disease. Meanwhile, most general hospitals only have patients' data on more frequently occurring diseases. In this example, the newly-found disease is a rare disease and general hospitals may share the same set of common diseases.

This extreme class imbalance can make learning these rare classes a difficult task. Again, I use the task of federated partially supervised multi-label classification to illustrate the problem of interest. Given a multi-label classification model, the representations extracted by the feature extractor are shared across common classes and rare classes. In this chapter, I observe an interesting phenomenon. Due to data decentralization and extreme class imbalance, improving the performance of rare classes

can inevitably decrease the performance of common classes, while maintaining the performance of common classes might lead to rare classes being completely ignored. Thus, the problem of interest in this chapter can be rephrased as how to leverage common classes to learn rare classes without impairing the performance of common classes in a federated system. Then, the key challenge of the problem of interest is how to mitigate the negative impact of *extreme* class imbalance between common classes and rare classes. In this chapter, another assumption is that data scarcity only occurs for rare classes, which is also the primary target. That is to say, common classes can have abundant partially labeled data or even unlabeled data. In contrast to the setup in Chapter 5, it might be worthwhile to use the unlabeled information to improve the performance.

To address the aforementioned challenges, a three-stage solution, `FedFew` is proposed. The core idea is to learn shared representations based on common classes and build a non-parametric classifier for rare classes. In the first stage, federated self-supervised learning is utilized to learn *class-agnostic* transferable representations as a strong initialization for the model weights of interest. In the second stage, an energy-based multi-label classifier aims to utilize the knowledge from the common classes to learn representations for the rare classes. In the third stage, a class-aware classification strategy is proposed to handle common classes and rare classes separately. Given only few partially labeled examples for the rare classes, I propose to use a *prototype*-based nearest-neighbor classifier for few-shot matching, instead of using a linear multi-label prediction head. To the best of my knowledge, this is the first study of federated partially supervised learning for the underrepresented classes.

The contributions of this chapter can be summarized as follows:

- I formulate and discuss for the first time the problem of federated partially supervised learning for the underrepresented classes.
- I propose a novel energy-based solution to federated partially supervised multi-label classification for rare classes, `FedFew`.
- I show initial evidence that federated partially supervised learning for rare classes is an under-explored problem compared with existing learning paradigms and provide an initial empirical understanding of the problem.

6.2 Related Work

6.2.1 Few-Shot Learning

In contrast to the concept of large-scale learning, *few-shot learning* is a new machine learning paradigm that learns from only limited labeled examples [178]. While deep learning models are commonly data-driven, data scarcity poses non-trivial challenges to practical tasks, especially when acquiring additional annotations is expensive, *e.g.* in the medical domain. A common setup in few-shot learning is q-shot learning. In the testing, a novel class unseen in the training set has only $q > 0$ labeled examples. As there are often multiple classes of interest involved in the training and testing, few-shot learning can also be formulated as a problem of multi-task learning (MTL) [17]. Under the interpretation of MTL, few-shot learning aims to learn shared representations across tasks (classes) in the training set than can generalize well on novel tasks unseen in the training set.

To achieve this goal, many efforts have been made to improve the performance of few-shot learning. Li *et al.* [55] first study the problem and adopts a Bayesian approach to utilize the knowledge learned from the training set. Instead, recent studies have a trend in formulating the training of few-shot learning as a *meta-learning* [78] problem, where the shared representations across tasks are meta-trained for unseen test classes. Koch *et al.* [99] use Siamese neural networks to rank the similarity between inputs. Santoro *et al.* [145] use external memory together with meta-learning. Ravi *et al.* [141] meta-learn an optimization algorithm for the learner. Many studies [172, 155, 159, 190] link meta-learning with distance metric learning and use nearest neighbor matching in the inference. Satorras *et al.* [148] further extend the idea of [172, 155, 159, 190] and use *graph neural networks* [66] (more precisely, *graph convolutional networks* [98]) to learn a distance metric.

Though few-shot learning shares a similar problem formulation interest in the chapter, *i.e.* only limited labeled examples are available for certain classes, and differences exist. First, few-shot learning or few-shot classification focuses on multi-class classification, where the problem of interest illustrated in this thesis is based on multi-label classification. Second, studies in few-shot learning do not consider the partial labels issue, which is the primary goal in this thesis. Third, few-shot learning tends to assume that the test classes are unseen in the training and the few labeled examples of test classes can only be accessed in the testing phase. Instead, this chapter assumes that the underrepresented classes have access to few labeled examples in the training. Fourth, existing few-shot learning studies usually comply with the n-way

q-shot setup in the training. Concretely, in the *episodic* training, n training classes are sampled and q labeled examples are sampled for each class to form a *support* set. Then, the model is trained to assign the correct label for a *query* example sampled from n selected classes. Under this setup, the total number of classes in the training set is commonly large. However, in the problem of interest, the classes of interest are pre-defined and should be the same between the training set and testing set. Fifth, few-shot learning focuses on the centralized training setup, while this chapter is an extension of federated partially supervised learning (Chapter 5).

6.2.2 Out-of-Distribution Detection

The definition of *out-of-distribution* arises from the situation that a machine learning model meets an example drastically different from the training data, *e.g.* the example belongs to a class unseen in the training set. For discriminative tasks such as classification, the model is trained to assign the correct label to the corresponding example in the training data. The model output can be completely meaningless if the input example does not belong to any of the classes seen during the training. Thus, out-of-distribution detection aims to determine whether an example is out-of-distribution.

A plethora of recent studies have tried to solve the issue of out-of-distribution detection. The mainstream solution is to design a measurement to quantify the uncertainty. For example, the common score-based methods include ODIN score [108, 79], confidence score based on *Mahalanobis* distribution [103], GradNorm score [82] based on gradient, and energy-based score [111, 173]. Another family of out-of-distribution detection methods is based on generative modeling, which estimates the in-distribution density and considers the low-density samples as out-of-distribution. Two state-of-the-art generative methods are likelihood ratio test [142] and likelihood regret [181]. It is also worth mentioning that self-supervised learning has also been shown the potential to improve out-of-distribution detection [76, 179]. This can be linked with Chapter 4.

This chapter is different from previous studies of out-of-distribution detection in many ways. First, the problem of interest is to tackle the extreme class imbalance, which, at its core, is not an out-of-distribution detection problem. Second, the majority of previous studies address the multi-class classification task and the current task of interest is partially supervised multi-label classification. Third, this chapter has a federated setup while the fore-mentioned studies are based on centralized training.

6.3 Problem Setup

Recall for problem setup of federated partially supervised learning in Section 5.3. The task of interest is federated partially supervised multi-label classification with C non-mutually exclusive classes of interest, where the decentralized partially labeled datasets are stored in a federated system with $K > 1$ clients. Let $\mathcal{D}_k$ denote the data stored in client $k \leq K$, $\mathcal{D}_k \cap \mathcal{D}_l = \emptyset$ for $k \neq l$ and $\{\mathcal{D}_k\}_{k=1}^K$ are non-IID data. For convenience, I 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. I also define $n_k^p = |\mathcal{P}_k|$ and $n_k^u = |\mathcal{U}_k|$.[1]

One can further assume that the classes of interest, denoted as $\mathcal{C}$, can be split into two mutually exclusive subsets, namely a set of underrepresented classes (denoted as $\mathcal{C}_u \subset \mathcal{C}$), which is also the primary target of this chapter, and a set of common classes (denoted as $\mathcal{C}_c = \mathcal{C} \setminus \mathcal{C}_u$). For simplicity, a representative case that there are $|\mathcal{C}_u| < K$ clients and each of these clients is annotated for only one underrepresented class is considered.[2] For the remaining $K - |\mathcal{C}_u|$ clients, one can assume that each client has *partially labeled* data for all common classes ($\mathcal{C}_c$).[3] It is additionally required to have $n_i^p \ll n_j^p \ \forall \ i \in \mathcal{C}_u, j \ \in \mathcal{C}_c$ to enforce the assumption of underrepresented classes. In the following context, I denote the clients with underrepresented classes as underrepresented clients and the clients with common classes as common clients for clarity. The learning outcome is to leverage the decentralized training data (both data on to train a multi-label classification model for $\mathcal{C}$. An intuitive example of the problem setup is illustrated in Figure 6.1.

The federated environment setup follows Section 5.3. In addition to K clients, there is a parameter server [104] for model aggregation. Let f_θ be the model of interest. In the parameter server, the parameter set θ_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 θ_k by training on $\mathcal{D}_k$ independently for a number of local epochs. Then, the parameter server aggregates $\{\theta_k\}_{k=1}^K$ collected from K clients to update θ_0. To ensure data government and comply with the data regulations in the medical domain [166, 53], I assume that the patient's data (either

[1] $|\cdot|$ is the cardinality of a set.

[2] 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 setup could be easily adapted to these extensions.

[3] Here, the examples with common classes are partially labeled with respect to the missing labels of the underrepresented classes, *i.e.* they are fully labeled if only common classes are considered. The assumption here is that the common classes are diseases with high prevalence, which can be easily collected and diagnosed; but the underrepresented classes are rare and only spotted in certain clients.

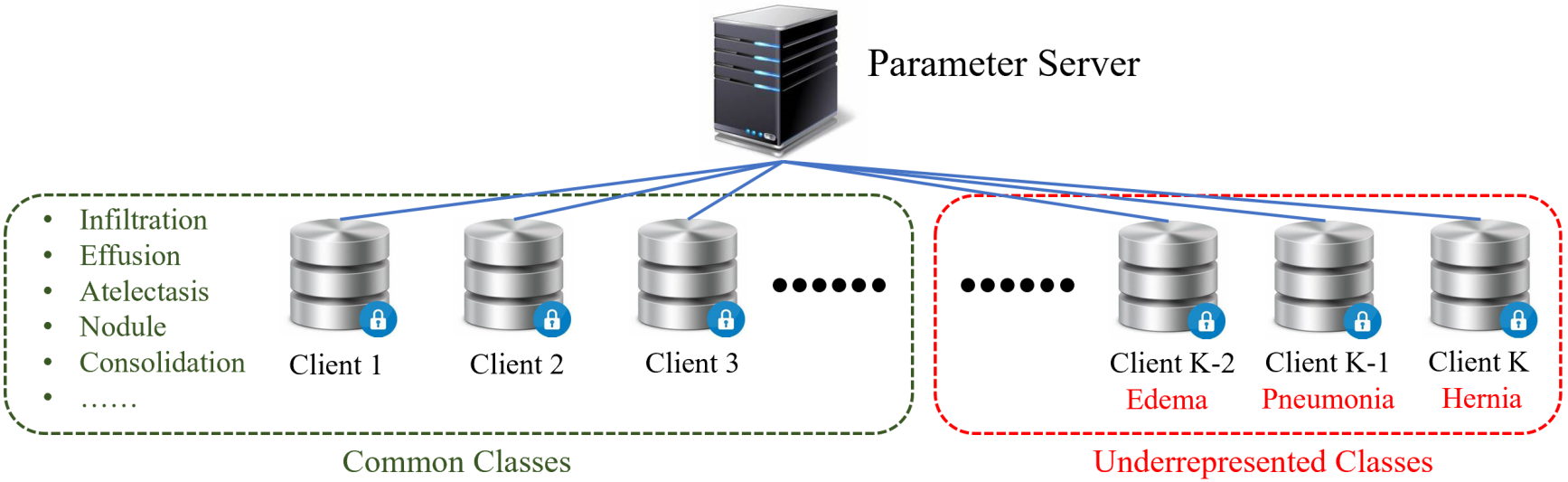


Figure 6.1: Illustration of federated partially supervised learning with rare classes (extreme class imbalance) for a multi-label classification task on chest X-ray images. In this example problem setup, the clients (data nodes) in the green dashed box are annotated for the same set of common classes and each client in the red dashed box are only annotated for one underrepresented class. From the perspective of a federated system with K clients, each example is partially labeled, *i.e.* either partially labeled for common classes or partially labeled for a single underrepresented class.

raw data or encoded data) in a client can not be uploaded to the parameter server 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 parameter server and the clients.

6.4 Preliminaries

6.4.1 Energy-Based Models

When it is difficult to keep track of all degrees of freedom of a large complex system in physics, physicists use *energy* to describe the system from a macroscopic view [93]. Inspired by this, machine learning scientists borrow this concept by using energy to describe a *state* in probabilistic modeling. In machine learning, energy-based models are used to capture the dependencies by associating an energy value (a scalar) to each possible state [102].

6.4.1.1 Energy Function

The core of the energy-based model is energy function. In predictive modeling, the learning outcome is to find an optimal energy function that assigns low energy values to the desired predictions (correct predictions) and high energy values to the undesired predictions (wrong predictions).

Mathematically, the goal of an energy-based model is to learn a mapping

$$E(x, y) : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R} \tag{6.1}$$

where $E(\cdot, \cdot)$ is the energy function of interest, x denotes a point in the input space, y is the corresponding prediction for $\boldsymbol{x}$, "$\times$" denotes the Cartesian product, and d denotes the dimension of the input.[4]

However, there is no such a *priori* guarantee that the energy scales are commensurate. It is convenient to link the energy function with modern machine learning models if the energy can be calibrated by mapping a collection of energy values into a collection of probability densities, whose sum is equal to 1. One way to achieve this is via the *Gibbs-Boltzmann distribution*. Let p denote the probability density function, by definition,

$$p(y|x) = \frac{e^{-\frac{E(x,y)}{\tau}}}{\int_{y'} e^{-\frac{E(x,y')}{\tau}}} \tag{6.2}$$

[4]Concretely, $E(\cdots)$ maps a state in a system to an energy value, where the state can have more than one observable variable. In this chapter, I simply use (x, y) is used to describe the state. When $\boldsymbol{x}$ is an image, $d = H \times W \times N_C$, where H, W, and N_C denote height, width, and number of channels of the image, respectively.

where $\int_{y'} e^{-\frac{E(x,y')}{\tau}}$ is also called the *partition function* and $\tau > 0$ is the temperature parameter.[5] By defining the $-T$ times of the log partition function

$$E(x) = -\tau \log \int_{y'} e^{-\frac{E(x,y')}{\tau}} \tag{6.3}$$

Equation (6.2) can be rewritten in a more elegant way:

$$p(y|x) = \frac{e^{-\frac{E(x,y)}{\tau}}}{e^{-\frac{E(x)}{\tau}}}. \tag{6.4}$$

Here, $E(x)$ is also known as *Helmholtz free energy* [116].

6.4.1.2 Connection with Machine Learning

With the probability density in Equation (6.4), I can formally study the relationship between the energy function and machine learning models. Assume there is a discriminative neural network classifier f, and the goal is to learn a mapping from the input to *logits*: $f(x) : \mathbb{R}^d \to \mathbb{R}^C$, where d and C are input and output dimensions for the neural network. Given a standard multi-class classification task with C classes, the logits can be used to derive the predicted probabilities with softmax function

$$p(y|x) = \frac{\frac{e^{f^y(x)}}{\tau}}{\sum_i^C \frac{e^{f^i(x))}}{\tau}} \tag{6.5}$$

where $f^y(x)$ is the logit of the y^{th} class label.

Equation (6.2) and Equation (6.5) share a similar form in the formula. Thus, the free energy $E(x; f)$ in Equation (6.4) can be expressed with the logits as

$$E(x; f) = -\tau \log \sum_i^C \exp^{\frac{f^i(x)}{\tau}}, \tag{6.6}$$

6.4.2 Distance Metrics

Here, the three most popular distance metrics used in few-shot learning are introduced. Given a feature extractor of interest f parameterized by θ, let $f_\theta(x)$ denote a feature representation of the input x, where $f_\theta(x)$ can be a feature vector or a set of feature maps. A distance metric d is used to measure the distance between the $f_\theta(x^s)$ and $f_\theta(x^q)$, where x^s denotes an input in the support set and x^q denotes a query input.

[5] In physics, the Gibbs-Boltzmann distribution describes the probability of finding a system in a certain state when the system is in thermal equilibrium with a heat bath at temperature T.

6.4.2.1 Cosine Distance

The cosine distance is defined as 1 minus the cosine similarity, which is the inner product between two feature *vectors*.

$$d_{CD}(a, b) = 1 - \frac{a}{\|a\|} \cdot \frac{b}{\|b\|} \tag{6.7}$$

In Equation (6.7), a and b are two inputs, *e.g.* $f_\theta(x^s)$ and $f_\theta(x^q)$, $\cdot$ denotes the inner product, and $\|\cdot\|$ denotes a norm operator. A constraint in Equation (6.7) is that the inner product requires two inputs to be two vectors and two vectors should have element-wise correspondence [43].

6.4.2.2 Squared Euclidean Distance

Two vectors can be represented as two points in a Cartesian system with corresponding dimensions. The squared Eculidean distance directly computes the squared length of a line segment between the two points by *Pythagorean* theorem.

$$d_{SED}(a, b) = \|a - b\|^2 \tag{6.8}$$

In Equation (6.8), a and b are two input feature vectors, *i.e.* $f_\theta(x^s)$ and $f_\theta(x^q)$.

6.4.2.3 Earth Mover's Distance

When the feature extractor is a convolutional neural network, the feature vector is acquired by applying a global average pooling operation on the feature maps extracted by the last convolutional layer. Thus, feature maps can also be used as representations in measuring the distance between two inputs [190, 110]. A common distance metric to measure the distance between two sets of feature maps is the *earth mover's* distance. Let $\boldsymbol{a}$ and $\boldsymbol{b}$ denote two feature maps, *i.e.* $\boldsymbol{a} = f_\theta(x^s)$ and $\boldsymbol{b} = f_\theta(x^q)$, and $\boldsymbol{a},\ \boldsymbol{b} \in \mathbb{R}^{H \times W \times M}$, where H, W, and M denote the dimensions of the feature maps, respectively. Now, $\boldsymbol{a}$ and $\boldsymbol{b}$ can be viewed as two sets of M-dimensional feature vectors, *i.e.* $\boldsymbol{a} = \{a_i | i = 1, 2, \cdots, HW\}$ and $\boldsymbol{b} = \{b_j | j = 1, 2, \cdots, HW\}$, where a_i and b_j are both feature vectors with a length of M, and i and j are the contiguous indexes of the positions of the feature maps. From the perspective of *optimal transport* [149], the similarity between $\boldsymbol{a}$ and $\boldsymbol{b}$ should be the minimum *transport cost* by transporting $\boldsymbol{a}$ to $\boldsymbol{b}$. The *transport ploytope* can be defined as

$$U(w^r, w^c) := \{\pi \in \mathbb{R}_+^{HW \times HW} | \pi \mathbb{1} = w^r, \pi^T \mathbb{1} = w^c\}, \tag{6.9}$$

where $\mathbb{1} = [\underbrace{1, \cdots, 1}_{HW}]$ are vectors of all ones, w^r and w^c are the marginal weights, *i.e.* marginal projections of π onto its rows and columns, respectively [6]. Following the *cross-reference mechanism*, the marginal weights are defined as

$$\begin{aligned} w_i^r &= \max\{a_i^T \cdot \text{GAP}(\boldsymbol{b}), 0\}, \\ w_j^c &= \max\{\text{GAP}(\boldsymbol{a})^T \cdot b_j, 0\}, \end{aligned} \tag{6.10}$$

where $\text{GAP}(\cdot)$ is the global average pooling operation that projects feature maps of shape $H \times W \times M$ into a feature vector of length M. The transport cost per unit between a_i and b_j is set as the cosine distance between them:

$$c_{ij} = 1 - \frac{a_i^T b_j}{\|a_i\| \|b_j\|}, \tag{6.11}$$

where $\|\cdot\|$ denotes the Euclidean norm.

With the transport plan π in Equation (6.9), the marginal weights in Equation (6.10), and the cost matrix $\boldsymbol{c}$ in Equation (6.11), the entropic optimal transport problem with a regularization entropy can be defined:

$$\min_{\pi \in U(w^r, w^c)} < \pi, \boldsymbol{c} > -\frac{1}{\lambda} H(\pi), \tag{6.12}$$

where $< \cdot, \cdot >$ stands for the *Frobenius* dot-product, $H(\cdot)$ is the entropy function and λ is a hyperparameter. Equation (6.12) can be solved via *Sinkhorn-Knopp algorithm* [29] with the optimal transport plan

$$\pi^* = \text{diag}(v) e^{-\lambda \boldsymbol{c}} \text{diag}(u), \tag{6.13}$$

where v and u are two non-negative vectors of scaling coefficients initialized as $\mathbb{1}$ and updated at $(t+1)^{\text{th}}$ iteration as

$$\begin{aligned} \forall i,\ v_i^{t+1} &\leftarrow \frac{w_i^r}{\sum_j e^{-\lambda \boldsymbol{c}} u_j^t} \\ \forall j,\ u_j^{t+1} &\leftarrow \frac{w_j^c}{\sum_i e^{-\lambda \boldsymbol{c}} v_i^{t+1}}. \end{aligned} \tag{6.14}$$

By repeating Equation (6.14) for a fixed number of times, the approximate optimal transport plan π^* can be obtained via Equation (6.13).

The earth mover's distance can be defined as

$$d_{EMD}(\boldsymbol{a}, \boldsymbol{b}) = 2 - 2 < \pi^*, 1 - \boldsymbol{c} > . \tag{6.15}$$

6.5 Method

This section introduces the proposed `FedFew` framework. Following the notations in Chapter 5, a multi-label classifier can be denoted as $g_\phi \circ f_\theta$, where f_θ is the feature extractor and g_ϕ is the multi-label predictor (prediction head). θ and ϕ are two sets of model weights. The federated setup can be found in Section 6.3.

6.5.1 Federated Self-Supervised Pre-Training

The first training stage is the federated self-supervised pre-training stage. The learning outcome of this stage is to pre-train the feature extractor f_θ to learn *class-agnostic* representations as a strong initialization for the next fine-tuning stage.

6.5.1.1 Motivation

In Section 5.6.5, the empirical results show that federated self-supervised learning does not help federated partially supervised learning. Then, why the first stage is needed?

Based on the discussion in Chapter 4, the task misalignment between the pretext and downstream tasks can negatively affect the downstream task performance. Note, in Chapter 5, an important assumption is the data scarcity, where each class of interest only has limited partial labels. This means that there might not be enough labels to fine-tune and mitigate the task misalignment.

However, in this chapter, common classes are assumed to have enough partial labels (*i.e.* they are unlabeled with respect to the rare classes) and additional unlabeled data are available. So, federated self-supervised pre-training can make a difference, at least on common classes. This will be demonstrated in Section 6.6.

6.5.1.2 Training

Theoretically, multiple existing self-supervised learning frameworks [73, 21, 67, 23] could serve as the local backbone. In this chapter, however, due to its lightweight nature, SimSiam [23] is leveraged for its simplicity. More details can be found in Section 4.3.3. Let θ_0^t denote the aggregated model weights in the PS at the end of the t^{th} training round. Thus, at the beginning of the $t+1^{\text{th}}$ round, the model weights at client k should be synchronized to θ_0^t. After the local updates of the $t+1^{\text{th}}$ round, the local model weights of client k are now θ_k^{t+1} and, θ_0^{t+1} is computed by applying

Algorithm 6.1 Federated self-supervised pre-training. "PS" denotes the parameter server. "Client_Update" is a function that performs local self-supervised training.

M: Metadata
T: Number of training rounds
Input: θ_0^0, $\{\mathcal{D}\}_{k=1}^K$
Output: θ_0^T

Initialize θ_0^0
for $t = 1, 2, \cdots, T$ **do**
  for $k = 1, 2, \cdots, K$ **do**
    $\theta_k^t \leftarrow \theta_0^{t-1}$ ▷ Synchronize with PS
    $\theta_k^t \leftarrow$ Client_Update(θ_k^t) ▷ Local self-supervised training
  Upload $\{\theta_k\}_{k=1}^K$ to PS
  $\theta_0^t \leftarrow \sum_{k=1}^K w_k^t \theta_k^t)$ ▷ Aggregate with Equation (5.6) and Equation (6.16)

Equation (5.6) on $\{\theta_k^{t+1}\}_{k=1}^K$, where

$$w_k = \frac{n_k^p + n_k^u}{\sum_{j=1}^K n_j^p + n_j^u}. \tag{6.16}$$

The complete pseudo-code is given in Algorithm 6.1.

6.5.2 Energy-Based Federated Learning with Partial Labels

In the second stage, the model initialized by the pre-trained weights in the first stage is fine-tuned by the available partial labels.

6.5.2.1 Motivation

In contrast to common classes, rare classes only have few partial labels. A standard supervised training process will be dominated by the common classes due to the class imbalance, *i.e.* the shared representations will be biased towards the common classes. Given a rare class $c \in \mathcal{C}_u$, even one can have a perfect linear weight g_ϕ^c for this rare class, $g_{\phi^c} \circ f_\theta$ can still tend to make wrong predictions due to the quality of the shared representations f_θ. The analysis in Section 5.4.3 still applies here.

While one can add more weights to the rare classes in the loss function Equation (3.4) or sampling, this will inevitably influence the performance of the common classes, reflected by the perturbations in the optimization of both f_θ and $g_{\phi^c} \forall c \in \mathcal{C}_c$. Thus, the goal is to utilize unlabeled information and maintain the performance of common classes at the same time, while learn shared representations that are meaningful to rare classes.

The proposed strategy is to divide the problem of interest into two sub-problems, following the *principle of divide and conquer*. Specifically, the first sub-problem is to successfully detect rare classes. For the first sub-problem, as the labeled examples for the rare classes are limited, training a supervised classifier is difficult. Instead, the sub-problem is reformulated as an energy-based out-of-distribution detection problem. Then, the second sub-problem is to handle common classes and rare classes separately.

6.5.2.2 Training

For standard multi-label classification, 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 rare classes, it is difficult to train a multi-label classification model for $\mathcal{C}$ or $\mathcal{C}_u$ directly. Instead, the first step is to train a multi-label classification model for common classes $\mathcal{C}_c$.

In contrast to previous studies [177, 140], the label information are encoded into a $(C_c + 1)$-dimensional vector, where $C_c = |\mathcal{C}_c|$. That is to say, there is an additional dimension denoted as the 0^{th} class in the $(C_c + 1)$-dimensional vector to specifically determine whether the input contains any common classes. The encoding rule is simple: when the additional dimension has the value of 0, the rest of C_c dimensions should have at least one value of 1; when the additional dimension has the value of 1, C_c dimensions should all be 0s. Note, this 0^{th} class only reflects the information on common classes, as label information regarding the rare classes for the examples partially labeled for the common classes is not available. The design of this 0^{th} class has an additional benefit as for this "new" class, the intra-class class distribution is more balanced, which can lead to more accurate prediction. Intuitively, examples with at least one positive common class will be labeled as "0" and examples without any positive common class will be labeled as "1".

Without loss of generality, I consider a client $k \in \mathcal{K}_c$ with only $\mathcal{C}_c$ labeled, where $\mathcal{K}_c$ denotes the clients with the common classes. As I am only interested in common classes now, the logit for the class c is

$$f_\Theta^c = g_{\phi^c} \circ f_\theta. \tag{6.17}$$

This informal reparameterization can simplify the mathematical notation and is well aligned with the energy function in Section 6.4.1.

Given an example x in client k with corresponding partial label y, the binary cross-entropy loss is

$$\mathcal{L}_{BCE}(x, y) = -\sum_{j \in \{0\} \cup \mathcal{C}_c} y_j \log(\text{sigmoid}(f_\Theta^j(x))) + (1 - y_j) \log(1 - \text{sigmoid}(f_\Theta^j(x))), \tag{6.18}$$

where $\text{sigmoid}(z) = \frac{1}{1+\exp^{-z}}$. From the perspective of multi-task learning, multi-label classification can be decomposed into multiple binary classification tasks. So, the free energy of x with respect to class j degenerates to

$$E(x; f_\Theta^j) = -\tau \log(1 + \exp^{\frac{f_\Theta^j(x)}{\tau}}), \tag{6.19}$$

which is a special case of Equation (6.6). Consequently, the *joint energy* [173] of x is then the sum of free energies over all common classes $\mathcal{C}_c$:

$$\begin{aligned} E(x, f_\Theta) &= \sum_{j \in \{0\} \cup \mathcal{C}_c} E(x; f_\Theta^j) \\ &= -\tau \sum_{j \in \{0\} \cup \mathcal{C}_c} \log(1 + \exp^{\frac{f_\Theta^j(x)}{\tau}}). \end{aligned} \tag{6.20}$$

A regularization term [111] is used to penalize the energy of x with a squared hinge loss:

$$\mathcal{L}_{E_c}(x) = \lambda \| \max(0, E(x; f_\theta) - m_c) \|_2^2, \tag{6.21}$$

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

$$\mathcal{L}_c = \mathcal{L}_{BCE}(x, y) + \mathcal{L}_{E_c}(x). \tag{6.22}$$

For clients with only rare classes (*i.e.* no common classes), only a regularization term is minimized:

$$\mathcal{L}_u = \lambda \| \max(0, m_u - E(x; f_\theta)) \|_2^2, \tag{6.23}$$

where the margin m_u is chosen empirically to increase the energy of x. Note, Equation (6.21) and Equation (6.23) are jointly designed to enlarge the *energy gap* between common classes $\mathcal{C}_c$ and rare classes $\mathcal{C}_u$. The models weights $\{\theta_k\}_{k=1}^K$ are aggregated via Equation (5.6), where

$$w_k = \frac{n_k^p}{\sum_{j=1}^K n_j^p}. \tag{6.24}$$

The complete pseudo-code is given in Algorithm 6.2.

Algorithm 6.2 Energy-based federated partially supervised training. "PS" denotes the parameter server. "Client_Update" is a function that performs local energy-based training.

M: Metadata
T_w: Number of warm-up training rounds
T: Number of training rounds
Input: Θ_0^0, $\{\mathcal{P}_k\}_{k=1}^K$
Output: Θ_0^T

for $t = 1, 2, \cdots, T_w$ **do** ▷ Warm up
  for $k \in \mathcal{K}_c$ **do**
    $\Theta_k^t \leftarrow \Theta_0^{t-1}$ ▷ Synchronize with PS
    $\Theta_k^t \leftarrow$ Client_Update(Θ_k^t) ▷ Equation (6.22)
  $\Theta_0^t \leftarrow \sum_{k \in \mathcal{K}_c} w_k^t \Theta_k^t$ ▷ Aggregate with Equation (5.6) and Equation (6.24)
for $t = T_w + 1, T_w + 2, \cdots, T$ **do**
  for $k = 1, 2, \cdots, K$ **do**
    $\Theta_k^t \leftarrow \Theta_0^{t-1}$ ▷ Synchronize with PS
    $\Theta_k^t \leftarrow$ Client_Update(Θ_k^t) ▷ Equation (6.22) or Equation (6.23)
  $\Theta_0^t \leftarrow \sum_{k=1}^K w_k^t \Theta_k^t$ ▷ Aggregate with Equation (5.6) and Equation (6.24)

6.5.3 Class-Aware Inference

The third stage is the inference stage. Following Section 6.5.2.1, two different methods are adopted for the inference of common classes and rare classes, respectively.

6.5.3.1 Inference for Common Classes

After the federated training in Section 6.5.2, f_Θ can be directly used as a multi-label classification model to predict common classes. The output is a $(C_c + 1)$-dimensional output vector. If the 0^{th} dimension of the output vector leads to a probability score greater than 0.5, *i.e.*

$$\text{sigmoid}(f_\Theta^0) \geq 0.5,$$

then all common classes are considered as negative. One can argue that there might be a conflict between the predictions between the 0^{th} dimension of the output vector and the C_c-dimensional vector. When the probability score of the 0^{th} class is greater than or equal to 0.5, the prediction of the 0^{th} class is prioritized.

If the 0^{th} class has a probability score less than 0.5, the C_c-dimensional vector is used as the final prediction for C_c common classes. In rare cases, one can have

$$\text{sigmoid}(f_\Theta^c) < 0.5 \; \forall c \in \mathcal{C}_c,$$

i.e. all common classes are predicted as negative though the 0^{th} class indicates there is at least common class positive. For such cases, all common classes are predicted as negative.

6.5.3.2 Inference for Rare Classes

Now, the energy defined in Equation (6.20) is used to detect the rare classes, *i.e.* if the energy of an example is lower than a threshold, then the example is deemed to contain no rare classes; if the energy is higher than the threshold, the test example is further matched to the nearest neighbor, given few partially labeled examples. Mathematically, let $\mathcal{F}_{OOD}$ denote a decision function for out-of-distribution detection.

$$\mathcal{F}_{OOD}(x, f_\Theta, \delta) = \begin{cases} 1 & \text{if } -E(x; f_\Theta^j) \geq \delta \\ 0 & \text{if } -E(x; f_\Theta^j) < \delta \end{cases} \tag{6.25}$$

The examples with higher energies are considered to contain rare classes and the examples with lower energies are considered to contain no rare classes. The negative energy is used to align with the standard definition for out-of-distribution score, *i.e.* higher score indicates the example is in-distribution [111]. The threshold δ is chosen to maximize the number of correctly classified training examples.

However, due to the constraint of data regulations, the training data and the test data are stored in separate clients. Thus, similar to [44], the metadata of rare classes are transferred to the parameter server at the final stage. Here, the metadata is the mean of the extracted features. For class $c \in \mathcal{C}_u$,

$$\boldsymbol{\mu}_c^{pos} = \frac{\sum_{i=1}^{n_c^{pos}} f_\theta(x_i^{pos})}{n_c^{pos}}, \tag{6.26}$$

$$\boldsymbol{\mu}_c^{neg} = \frac{\sum_{i=1}^{n_c^{neg}} f_\theta(x_i^{neg})}{n_c^{neg}}, \tag{6.27}$$

where I use *pos* and *neg* to denote the positive and negative examples of the rare class c, respectively. Note, $\boldsymbol{\mu}_c^{pos}$ and $\boldsymbol{\mu}_c^{neg}$ factually define the *prototypes* in the few-shot learning literature [155, 46]. With the *dual-prototypes* for the rare class c, the test example is matched to the closer one by computing the distance between the features of test example and the two prototypes. Let $\mathcal{F}_{NN}$ denotes a decision function for nearest-neighbor classification.

$$\mathcal{F}_{NN}(x, f_\theta, d, \boldsymbol{\mu}_c^{pos}, \boldsymbol{\mu}_c^{neg}) = \begin{cases} 1 & \text{if } d(f_\theta(x), \boldsymbol{\mu}_c^{pos}) \geq d(f_\theta(x), \boldsymbol{\mu}_c^{neg}) \\ 0 & \text{if } d(f_\theta(x), \boldsymbol{\mu}_c^{pos}) < d(f_\theta(x), \boldsymbol{\mu}_c^{neg}) \end{cases} \tag{6.28}$$

$\mathcal{F}_{NN}$ outputs "1" for positive prediction of the rare class c and "0" for negative prediction of the rare class c. Three common distance metrics used in few-shot learning are introduced in Section 6.4.2.

6.5.3.3 Remarks

Note, the prototype-based nearest neighbor classifier usually performs worse than the linear classifier when there are enough labeled data. A rule of thumb in few-shot learning commonly consider $n^{pos} \leq 10$. For $n^{pos} > 10$, k-nearest neighbors algorithm might be helpful, especially when n^{neg} is large.

It is possible that there are multiple prototypes collected from different clients for the rare class c. Under this case, majority voting (*i.e.* k-nearest neighbors algorithm) is adopted when there are multiple positive and negative prototypes.

Again, the goal of this chapter is not complete privacy-preserving, an integration of differential privacy [51] techniques can be integrated with model weights and metadata. This is left as future work.

6.5.4 Theoretical Analysis

Recall that in Section 6.5.2.2, a special 0^{th} class is designed such that an example is labeled as "1" when it contains no common classes and labeled as "0" when it contains at least one common class. There is a hidden advantage for this design is energy-based learning.

Given an example x, $E(x, f^j_\Theta)$ in Equation (6.19) will be smaller when the class j is positive than its counterpart when the class j is negative as $f^j_\Theta(x)$ will be higher in the former case. To better understand the problem, I assume that there is a *oracle* function $f^j_{\Theta^*}$. It has the following properties: when x is positive for class j, $f^j_{\Theta^*} \to \infty$; when x is negative for class j, $f^j_{\Theta^*} \to -\infty$.[6] Under this assumption, if x^* denotes an arbitrary example that has no common classes positive, for the joint energy in Equation (6.20),

$$E(x^*, f^*_\Theta) == -\tau \sum_{j \in \mathcal{C}_c} \log(1 + \exp^{\frac{f^j_{\Theta^*}(x^*)}{\tau}}) \to 0. \tag{6.29}$$

If $x^\dagger$ denotes an arbitrary example that has at least one common class positive,

$$E(x^\dagger, f^*_\Theta) = -\tau \sum_{j \in \mathcal{C}_c} \log(1 + \exp^{\frac{f^j_{\Theta^*}(x^\dagger)}{\tau}}) \to -\infty. \tag{6.30}$$

Note, $E(x^*, f^*_\Theta)$ has reached the upper bound (*i.e.* 0 is the upper bound). As the joint energy can not be larger than 0, it is impossible to find a suitable threshold for out-of-distribution detection.

[6] Or, equivalently, when x is positive for class j, the probability score $\text{sigmoid}(f^j_{\Theta^*}(x)) \to 1$; when x is negative for class j, the probability score $\text{sigmoid}(f^j_{\Theta^*}(x)) \to 0$.

The above derivation is repeated. But this time, the design of the 0^{th} dimension is included.

$$E(x^*, f^*_\Theta) == -\tau \sum_{j \in \{0\} \cup \mathcal{C}_c} \log(1 + \exp \frac{f^j_{\Theta^*}(x^*)}{\tau}) \to -\infty \tag{6.31}$$

and

$$E(x^\dagger, f^*_\Theta) = -\tau \sum_{j \in \{0\} \cup \mathcal{C}_c} \log(1 + \exp \frac{f^j_{\Theta^*}(x^\dagger)}{\tau}) \to -\infty. \tag{6.32}$$

It is much easier to find a suitable threshold for out-of-distribution detection.

The above analysis is hypothetical and it is difficult to train such an *oracle* function in practice. But the above analysis suggests that including a new class can efficiently enlarge the *buffer* from which the suitable threshold can be selected.

6.6 Experiments

6.6.1 Experimental Setup

To provide empirical insights into the problem and ensure fair comparisons with the baselines, the same experimental setups are shared across all experiments in this chapter.

6.6.1.1 Data

The multi-label dataset ChestX-ray14 [177] is used in this chapter. I adopt its default batch splits to ensure reproducibility. There are fourteen non-mutually exclusive thoracic diseases in the dataset, *i.e.* a patient can be diagnosed with multiple diseases. Based on the label statistics of the dataset, *edema*, *pneumonia*, and *hernia* are chosen as three rare classes and the remaining eleven classes are considered as common classes. Note, most chest X-ray images do not contain any diseases.

In addition to the multi-label classification dataset, another chest X-ray dataset is prepared to perform ImageNet-style [34] self-supervised learning evaluation. Here, I use a public multi-class pneumonia dataset [90]. There are a total of 5856 chest X-ray images collected from children, including 2780 images labeled as *bacterial pneumonia*, 1493 images labeled as *viral pneumonia*, and 1583 images labeled as *normal*.

6.6.1.2 Simulation

I use 6 batches to simulate $K = 6$ clients, where the first three batches are used to simulate the partially labeled datasets for common classes, where $n^p_k = 5{\times}10^3$ images

are randomly sampled to keep partial labels for the common classes.[7] Each of the remaining three batches contains one of the three rare classes. 10 positive examples and 90 negative examples are randomly sampled to simulate the class imbalance for rare classes, *i.e.* $n_k^p = 100$. See Figure 6.1 for an illustration of the class assignment. From the remaining batches, 100 positive and 100 negative examples are held out for each class as the test set.

6.6.2 Implementation

6.6.2.1 Data Pre-Processing

Each chest X-ray image has a resolution of 1024×1024. During both training and testing of the model, each chest X-ray image is first resized to 224×224 and the image is then normalized by instance normalization [30] in Equation (3.6). The data augmentation policy proposed in Listing 4.1 is used for pre-training.

6.6.2.2 Network Architecture

Two network backbones for the feature extractor f_θ are explored, which are ResNet34 [74] and DenseNet121 [81]. These two models are lightweight models that are commonly used in federated learning. Following Chapter 4 and Chapter 5, SimSiam [23], a state-of-the-art self-supervised learning framework, is used to pre-train f_θ locally.

Each of the federated methods has $K + 1$ copies of models for K clients and the parameter server. All models are implemented in PyTorch [132] on an NVIDIA Tesla V100.

6.6.2.3 Training

For both self-supervised pre-training and energy-based learning, the local optimization is performed by a standard Adam [97] optimizer with fixed learning rate 10^{-3} and batch-size 64. For simplicity, I set $\tau = 1$ when computing the energy, following [111]. The values of m_c and m_u can be chosen around the mean of energy scores from a model trained without energy-based loss (*e.g.* warm-up stage in this chapter) for images from the clients of common classes and the clients of rare classes, respectively. In this chapter, I simply set $m_c = -5$ and $m_u = -25$ to intentionally enlarge the energy gap and I use $\lambda = 0.01$.

The synchronization and aggregation for federated methods are performed every 10 epochs. For the second stage, I set $T_w = 20$ and $T = 100$.

[7] I use batches 2 to 7. Each batch has 10^4 chest X-ray images and similar label distributions.

6.6.2.4 Evaluation Metrics

In this chapter, different evaluation metrics are adopted for different downstream tasks. In Section 6.6.3.1, since the evaluation is proxy-based, I simply use accuracy for the multi-class classification task. In Section 6.6.3.2, for rare classes, I mainly use F1 score to study the class-wise performance (see Section 2.3.2.1 for details of F1 score), and for common classes, I use AUROC to check the overall performance (see Section 2.3.2.2 for details of AUROC). The reasons behind the choice are explained in Section 2.3.2.3. In Section 6.6.3.4, as the evaluation is still for rare classes, I also use F1 score.

6.6.3 Empirical Analysis

6.6.3.1 Impact of Federated Self-Supervised Pre-Training

In Chapter 5, the initial empirical results show that federated self-supervised pre-training does not necessarily help in federated partially supervised learning, especially when each client has annotations for a different set of classes. The first step is to evaluate the impact of federated self-supervised pre-training.

Following previous self-supervised learning studies, the performance of federated self-supervised learning performance is examined via the *linear classification protocol* [73, 21, 67].[8] Similar to [44], all the weights of f_θ are frozen except the last layer and only the last layer is fine-tune on a public pneumonia dataset. In this dataset, there are three mutually exclusive classes, *normal*, *bacteria pneumonia*, and *virus pneumonia*. The images of each class are randomly split into two halves as the training and test sets. I use the test accuracy as the *proxy* measure to assess the representation learning performance.

Firstly, I provide a counter-intuitive observation that more disease images might not improve the performance. Two clients are created with Chest-Xray14 data, where one client contains 10^4 images without any diseases and the other client contains $r\times10^4$ images with various diseases. The results in Table 6.1 show that FedAvg [115] 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. This suggests the potential of using large-scale unlabeled images in the pre-training as most of them contain no diseases at all.

Secondly, I examine the impact of the training epochs for federated self-supervised pre-training in Figure 6.2, where f_θ is pre-trained on the 6 clients described in

[8] See Section 4.7.2.5 for more information.

Table 6.1: Impact of class imbalance on federated self-supervised pre-training. The mean accuracy over three random seeds are reported. For ResNet34, $r = 0.5$ gives the highest accuracy; for DenseNet121, $r = 0.1$ gives the highest accuracy. Overall, a large r (*e.g.* $r = 1$) might not help, *i.e.* it is unnecessary to collect a large number of images with diseases.

Epoch	$r = 1$		$r = 0.5$		$r = 0.1$	
	ResNet34	DenseNet121	ResNet34	DenseNet121	ResNet34	DenseNet121
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

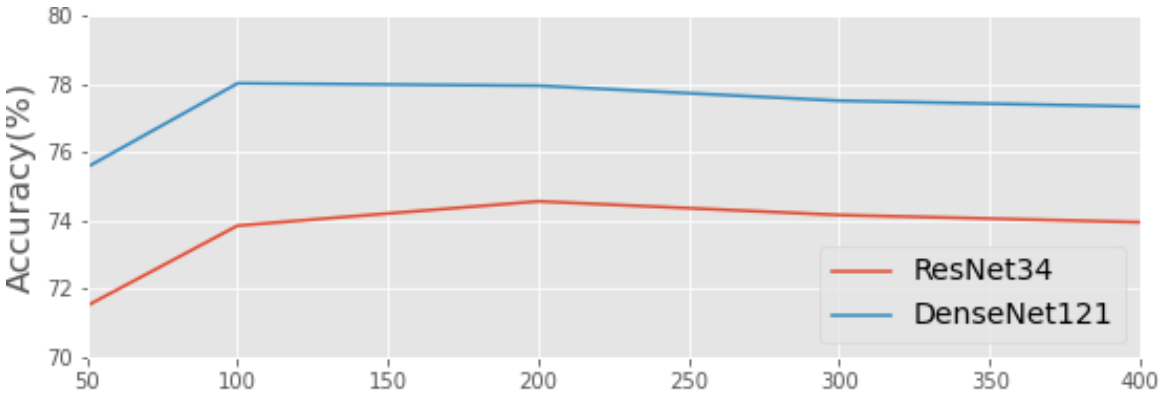


Figure 6.2: Impact of the number of training epochs on federated self-supervised pre-training. More epochs will not lead to diminishing performance gain but decreasing results.

Section 6.6.1.2. In contrast to the empirical findings collected from general images [73, 21, 67, 23], more epochs will not lead to diminishing performance gain but decreasing results. In the following experiments, I will use DenseNet121 as the default network because it has better performance than ResNet34 and I will use the pre-trained weights with 100 epochs because it leads to the highest linear classification performance.

6.6.3.2 Evaluation of Energy-Based Federated Training

Following the experimental setup in Section 6.6.1, the proposed framework is evaluated against a few seminal baselines from two aspects, *i.e.* the goal is to achieve high accuracy on rare classes while maintaining robust performance on common classes.

The first baseline is a standard multi-label classification model [140], which is trained with FedAvg on decentralized partially labeled data of all 14 classes and weighted binary cross-entropy [140]. I simply use *MLC w/o FSSL* and *MLC w/ FSSL* to differentiate whether f_θ is pre-trained with federated self-supervised learn-

ing. The second baseline is a hypothetical local nearest-neighbor classifier (*NN*) that does not use prototypes, where f_θ is either pre-trained with federated self-supervised learning alone or further fine-tuned with *MLC w/ FSSL*. Note, *NN* can not be used in a federated system due to data regulations. For FedFew, I use *EBM* to denote whether the energy-based loss is included in the training. The distance metric in this experiment is cosine distance. I choose cosine distance because of its robust performance in Section 6.6.3.4. The results on rare classes are presented in Table 6.2 and include the mean accuracy (A), precision (P), recall (R), and F-1 (F) score over three random seeds. Note, standard multi-label classification models fail to detect any images of the rare classes due to extreme class imbalance. With only prototypes (as only metadata can be transferred to the parameter server), 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 6.2, I also report the mean AUROC for the eleven common classes over three runs in Table 6.3. I find that federated self-supervised pre-training can boost the performance for standard MLC. While FedFew (*w/o EBM*) and FedFew (*w/ EBM*) both outperform *MLC w/ FSSL*, including *EBM* slightly improves the performance on common classes. I conjecture that *EBM* can play a role as regularization in the federated training. By combining the results from both Table 6.2 and Table 6.3, I conclude that FedFew (*w/ EBM*) provides a robust solution to the problem of interest.

6.6.3.3 Analysis on Energy

As an ablation study, I further design an experiment to demonstrate the energy-based learning outcome. Two DenseNet121s are trained without and with *EBM* following the setup in Section 6.6.3.2. For a fair comparison, two models are trained with twelve classes (*i.e.* with the 0^{th} class). The density plots of learned energies for images in the clients with rare classes are visualized in Figure 6.3 and Figure 6.4. Here, the energies are computed based on Equation (6.1), where the models have converged. The images with partial labels of rare classes are simply denoted as "positive" and the images without partial labels of rare classes are denoted as "negative". As shown in Figure 6.3, without *EBM*, the densities of positive and negative examples significantly overlap, *i.e.* the *energy gap* is small. This situation is largely alleviated with *EBM*, shown in Figure 6.4. This suggests that the proposed energy-based learning not

Table 6.2: Performance comparison on the rare classes. The standard MLC and NN models fail to predict the rare classes. A, P, R, and F denote the mean accuracy, precision, recall, and F1-score over three random seeds, respectively. Blue denotes the highest number in each column.

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	0.00 ± 0.00	0.00 ± 0.00	0.50 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.50 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
MLC w/ FSSL	0.50 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.50 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.50 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
NN (FSSL)	0.53 ± 0.00	0.53 ± 0.00	0.50 ± 0.00	0.52 ± 0.00	0.51 ± 0.00	0.51 ± 0.00	0.46 ± 0.00	0.48 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.54 ± 0.00	0.52 ± 0.00
NN (MLC w/ FSSL)	0.53 ± 0.00	0.53 ± 0.00	0.50 ± 0.00	0.52 ± 0.00	0.51 ± 0.00	0.51 ± 0.00	0.46 ± 0.00	0.48 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.54 ± 0.00	0.52 ± 0.00
FedFew w/o EBM	0.71 ± 0.00	0.85 ± 0.00	0.50 ± 0.00	0.63 ± 0.00	0.69 ± 0.00	0.84 ± 0.00	0.46 ± 0.00	0.59 ± 0.00	0.72 ± 0.00	0.83 ± 0.00	0.54 ± 0.00	0.65 ± 0.00
FedFew w/ EBM	0.75 ± 0.00	1.00 ± 0.00	0.50 ± 0.00	0.67 ± 0.00	0.73 ± 0.00	1.00 ± 0.00	0.46 ± 0.00	0.63 ± 0.00	0.77 ± 0.00	1.00 ± 0.00	0.54 ± 0.00	0.70 ± 0.00

Table 6.3: Performance comparison on the common classes. AUROC denotes the mean AUROC of eleven common classes over three random seeds. Following [140], only the mean AUROC is reported. Blue denotes the highest number.

Method	AUROC
MLC w/o FSSL	0.6905
MLC w/ FSSL	0.7227
FedFew w/o EBM	0.7423
FedFew w/ EBM	0.7479

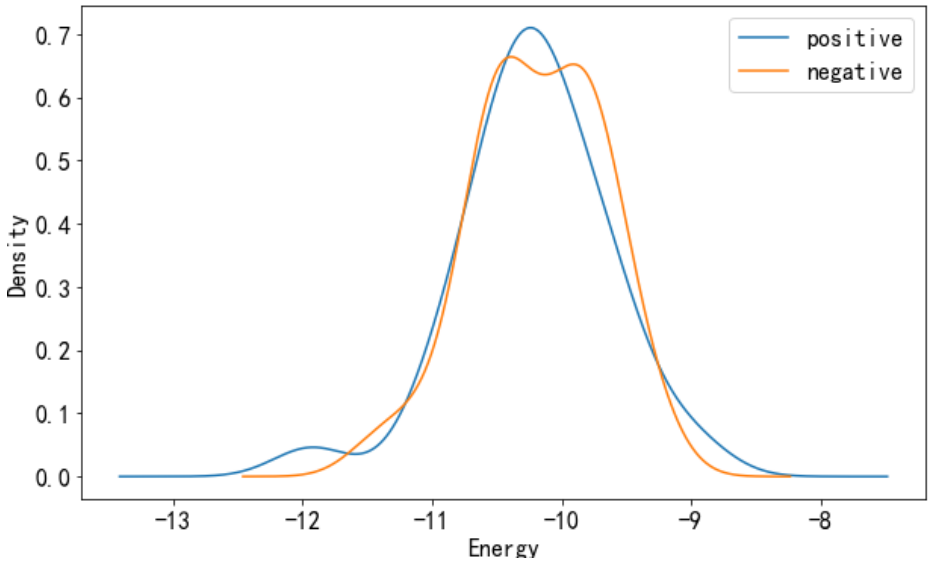


Figure 6.3: Energy gaps between positive and negative cases in clients with rare classes without *EBM* in federated training. Under a standard federated training procedure, the energy distributions of positive and negative cases significantly overlap. It is difficult to differentiate the positive cases from the negative cases.

only leads to improved performance, but also efficiently enlarges the energy gap for out-of-distribution detection.

6.6.3.4 Choice of Distance Metric

I consider three distance metrics introduced in Section 6.4.2 for the nearest-neighbor matching is Section 6.5.3.2, which are cosine distance [172], Euclidean distance [155], and earth mover's distance [190]. The average F-1 scores for three distance metrics are presented in Table 6.4. The cosine distance is chosen due to its robustness shown in Table 6.4, where the cosine distance leads to the highest F1-score.

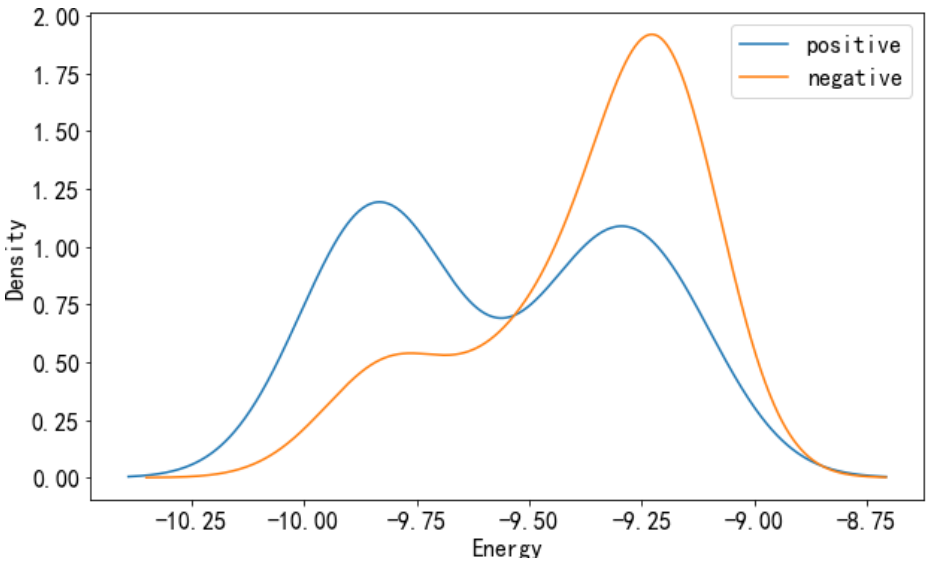


Figure 6.4: Energy gaps between positive and negative cases in clients with rare classes with *EBM* in federated training. Under the proposed energy-based federated training procedure, the energy distributions of positive and negative cases show clear difference in contrast to Figure 6.3.

Table 6.4: Performance comparison on the choice of distance metrics. F1 denotes the mean F1-score of three rare classes over three random seeds. Blue denotes the highest number. The cosine distance leads to the highest F1-score.

Distance	F1
Cosine Distance	0.67 ± 0.00
Euclidean Distance	0.60 ± 0.00
Earth Mover's Distance	0.43 ± 0.00

6.7 Conclusion

In this chapter, I raise awareness of an under-explored problem in federated partially supervised learning, namely the learning of rare or underrepresented classes from decentralized partially labeled medical images. I not only provide a novel energy-based solution to this challenging problem but also provide the first empirical understanding of federated partially supervised learning with extreme class imbalance. This chapter poses a new research direction on label-efficient learning.

Chapter 7

Conclusion

7.1 Summary

7.1.1 Contributions

In this thesis, I formalize the learning paradigm of partially supervised learning for the first time and review the recent progress on this emerging research question. To answer the four research questions in Section 1.2, I explore four new research topics of partially supervised learning: partially supervised learning on small-scale data, partially supervised learning with unlabeled data, federated partially supervised learning, and federated partially supervised learning with extreme cross-site class imbalance. I provide both theoretical and empirical understandings of these new topics and propose novel solutions for each. Specifically, in Chapter 3, I propose a data augmentation strategy based on vicinal risk minimization to mitigate data scarcity. In Chapter 4, I propose a pretext task based on vicinal risk minimization to utilize unlabeled data. In Chapter 5, I propose a federated learning framework based on meta-learning to address data decentralization for PSL. In Chapter 6, I propose an energy-based federated learning pipeline to handle underrepresented classes.

In addition to partially supervised multi-label classification, the theoretical and empirical results presented in this thesis can be generalized to more complex machine learning tasks. For example, in [39], I leverage vicinal risk minimization and proposed a partially supervised framework for medical image segmentation. The idea of self-supervised pre-training and federated partially supervised learning can also be extended to other machine learning tasks in a similar fashion [39]. These are left as future works.

By formulating four new research problems, this thesis raises awareness that partially supervised learning is still an under-explored research field. A list of potential

topics is provided in Section 7.2.

In summary, this thesis provides a comprehensive view of a new research field and raises awareness of several interesting topics.

7.1.2 Limitations

As the first systematic study on partially supervised learning, this thesis has several limitations.

Firstly, as the first exploratory work on the field, I focus more on the learning paradigm, *i.e.* problem formulation, instead of a specific practical task. To conduct theoretical and experimental analysis, the assumptions and experimental setups are simplified. Though the results have offered insights into understanding the problem of interest, the application of the proposed methods to practical problem still requires further attention. Additional challenging conditions shall be considered in a realistic environment. Besides, I only use multi-label classification as the base task to illustrate the concept. There are more complex tasks to be solved in the field of partially supervised learning. The adaptation of the results from multi-label classification to more complex tasks also need further work.

Secondly, during my DPhil study, the computational resource is one of the bottlenecks. I only have limited access to GPU computing resource. This limits the scale of the simulated experiments, *e.g.* the number of classes of interest and the number of clients in a federated system. Theoretically, the methods proposed are designed to be scalable. However, the large-scale evaluation is more like an engineering question. Again, the practical applications will require case-by-case discussion.

Thirdly, I only discuss four unexplored topics. My DPhil study significantly overlaps with the COVID-19 pandemic. My time and energy were inevitably influenced by the infection in a negative fashion. However. there are potentially more unexplored topics to be formulated and discussed. These are also left as future work.

7.2 Future Directions

The results presented in the thesis also provide insights for future directions.

Task Domain The focus of this thesis is multi-label classification on chest X-ray images. To address this specific task, the proposed techniques aim to leverage the unique characteristics associated with it. However, it is important to explore different task domains as well. In the field of machine learning, there are other tasks that

also involve a multi-task learning formulation, such as object detection and semantic segmentation. For instance, in a recent study by [39], partially supervised semantic segmentation on medical images was formulated. Moreover, there are other medical data modalities commonly used in medical imaging, such as computed tomography and magnetic resonance imaging. Dong *et al.* [40] utilize the sequential order to create a pretext task for self-supervised learning. Apart from medical tasks, there are also challenging tasks that can benefit from partial supervision in various application areas, including finance, internet of things, science, and security, as long as the cost of data annotation is non-trivial. Therefore, it is crucial to invest more efforts in understanding these emerging task domains and designing appropriate solutions.

Class Imbalance Although Chapter 3, Chapter 5, and Chapter 6 address the issue of class imbalance in partially supervised learning and present initial findings, real-life class imbalance challenges are more intricate and demanding than the examples illustrated in this thesis. The thesis merely provides a few toy examples to facilitate empirical comprehension. Consequently, there exist more formidable scenarios in the real world. For instance, there may be multiple sub-tasks of interest, and the task distributions could exhibit a "long-tailed" pattern. Furthermore, these data challenges should be closely connected to specific application scenarios. It is essential to comprehend these practical data challenges and resolve real-life problems effectively.

Domain Shift In this thesis, the issue of domain shift is only illustrated through experiments in Chapter 5, where different clients are considered as distinct source domains. Similar to class imbalance, domain shift is an important topic in machine learning, specifically in the field of *domain adaptation* [175]. Since partially labeled datasets are often collected from multiple sources, it is natural to associate the problem of interest with *multi-source domain adaptation* [158]. In other words, domain adaptation techniques should be adapted for partially supervised learning, and potentially new algorithms can be designed specifically for this task.

Federated System Chapter 5 and Chapter 6 discuss the basic scenarios of federated partially supervised learning. However, this thesis only utilizes FedAVG [115], which is a fundamental federated setup, to illustrate the concept of federated partially supervised learning. In terms of system design, advanced federated learning algorithms can be developed to reflect the recent advances in federated systems [105]. Additionally, to enable the practical application of federated partially supervised

learning, a deeper empirical understanding of this new topic is required from an engineering perspective. For instance, apart from simulation, factors such as hardware capabilities and internet communication need to be taken into account.

7.3 Concluding Remarks

Throughout this thesis, partially supervised learning has demonstrated significant theoretical and practical value to the machine learning community. Despite the progress made, numerous challenges persist (see Section 2.2.2). In addition to the data challenges, there are still crucial questions in terms of learning awaiting resolution. One prominent issue is the design of effective strategies for utilizing informative partially labeled and unlabeled data, which can greatly impact learning performance. Additionally, the development of robust algorithms capable of handling complex and diverse datasets remains an active area of research. The field of partially supervised learning also poses important questions regarding the generalization capabilities of models trained under limited supervision and their potential biases.

Addressing these challenges requires persistent effort and meticulous investigation. Large-scale experiments must be conducted, diverse datasets must be gathered, and interdisciplinary collaborations must be engaged to ensure the reliability and validity of theoretical and empirical findings. Furthermore, the exploration of new mathematical frameworks, such as graph-based approaches and diffusion models [27], as well as large language and vision models [137], could provide promising avenues for advancing partially supervised learning.

The ultimate goal of this arduous journey is to create machine learning systems that have a profound and positive impact on society as a whole. Accurately predicting diseases or diagnosing conditions can save lives and improve the quality of patient care. This is just a simple example that illustrates the immense potential of this research domain. Imagine the potential applications in autonomous driving, where robust partially supervised learning algorithms could enhance safety and efficiency, revolutionizing transportation. Similarly, the finance and insurance industries could also benefit from partially supervised learning to reduce financial risks. Advancing from the exploratory groundwork laid out in this thesis to a machine learning system with far-reaching impacts on human society will require unwavering dedication and rigorous research.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[3] Haftay Gebreslasie Abreha, Mohammad Hayajneh, and Mohamed Adel Serhani. Federated learning in edge computing: a systematic survey. *Sensors*, 22(2):450, 2022.

[4] G Anandalingam and Terry L Friesz. Hierarchical optimization: An introduction. *Annals of Operations Research*, 34(1):1–11, 1992.

[5] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*, volume 9. Cambridge University Press, 1999.

[6] YM Asano, C Rupprecht, and A Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020.

[7] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[8] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15535–15545, 2019.

[9] Pierre Baldi and Fernando Pineda. Contrastive learning and neural oscillations. *Neural Computation*, 3(4):526–545, 1991.

[10] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

[11] Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4):719–760, 2020.

[12] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010.

[13] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 137–144, 2006.

[14] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[15] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, pages 41–48, 2009.

[16] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546—-2554, 2011.

[17] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[18] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[19] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Advances in Neural Information Processing Systems*, pages 416–422, 2001.

[20] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[21] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[22] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[23] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[24] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in Neural information Processing Systems*, pages 577–585, 2015.

[25] Muhammad EH Chowdhury, Tawsifur Rahman, Amith Khandakar, Rashid Mazhar, Muhammad Abdul Kadir, Zaid Bin Mahbub, Khandakar Reajul Islam, Muhammad Salman Khan, Atif Iqbal, Nasser Al Emadi, et al. Can ai help in screening viral and covid-19 pneumonia? *IEEE Access*, 8:132665–132676, 2020.

[26] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, 2007.

[27] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[28] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.

[29] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, pages 2292–2300, 2013.

[30] Wei Dai, Nanqing Dong, Zeya Wang, Xiaodan Liang, Hao Zhang, and Eric P Xing. Scan: Structure correcting adversarial network for organ segmentation in chest x-rays. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 263–273. Springer, 2018.

[31] Wei Dai, Yi Zhou, Nanqing Dong, Hao Zhang, and Eric Xing. Toward understanding the impact of staleness in distributed machine learning. In *International Conference on Learning Representations*, 2019.

[32] Ittai Dayan, Holger R Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, Anas Z Abidin, Andrew Liu, Anthony Beardsworth Costa, Bradford J Wood, Chien-Sung Tsai, et al. Federated learning for predicting clinical outcomes in patients with covid-19. *Nature Medicine*, 27(10):1735–1743, 2021.

[33] Virginia R de Sa. Learning classification with unlabeled data. In *Advances in Neural Information Processing Systems*, pages 112–119, 1994.

[34] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[35] Thomas G Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15. Springer, 2000.

[36] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

[37] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017.

[38] Nanqing Dong, Michael Kampffmeyer, Xiaodan Liang, Zeya Wang, Wei Dai, and Eric Xing. Unsupervised domain adaptation for automatic estimation of cardiothoracic ratio. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 544–552, 2018.

[39] Nanqing Dong, Michael Kampffmeyer, Xiaodan Liang, Min Xu, Irina Voiculescu, and Eric Xing. Towards robust partially supervised multi-structure

medical image segmentation on small-scale data. *Applied Soft Computing*, page 108074, 2022.

[40] Nanqing Dong, Michael Kampffmeyer, and Irina Voiculescu. Self-supervised multi-task representation learning for sequential medical images. In *European Conference on Machine Learning*, pages 779–794. Springer, 2021.

[41] Nanqing Dong, Michael Kampffmeyer, and Irina Voiculescu. Learning underrepresented classes from decentralized partially labeled medical images. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 67–76. Springer, 2022.

[42] Nanqing Dong, Michael Kampffmeyer, Irina Voiculescu, and Eric Xing. Federated partially supervised learning with limited decentralized medical images. *IEEE Transactions on Medical Imaging*, 2022.

[43] Nanqing Dong, Matteo Maggioni, Yongxin Yang, Eduardo Pérez-Pellitero, Ales Leonardis, and Steven McDonagh. Residual contrastive learning for image reconstruction: Learning transferable representations from noisy images. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2930–2936, 2022.

[44] Nanqing Dong and Irina Voiculescu. Federated contrastive learning for decentralized unlabeled medical images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 378–387. Springer, 2021.

[45] Nanqing Dong, Jiayi Wang, and Irina Voiculescu. Revisiting vicinal risk minimization for partially supervised multi-label classification under data scarcity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4212–4220, 2022.

[46] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *British Machine Vision Conference*, 2018.

[47] Nanqing Dong, Min Xu, Xiaodan Liang, Yiliang Jiang, Wei Dai, and Eric Xing. Neural architecture search for adversarial medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 828–836. Springer, 2019.

[48] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[49] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 766–774, 2014.

[50] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep convnet for multi-label classification with partial labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 647–657, 2019.

[51] Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12. Springer, 2006.

[52] Thibaud Ehret, Axel Davy, Pablo Arias, and Gabriele Facciolo. Joint demosaicking and denoising by fine-tuning of bursts of raw images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8868–8877, 2019.

[53] European Commission. General data protection regulation, 2016.

[54] Xi Fang and Pingkun Yan. Multi-organ segmentation over partially labeled datasets with multi-scale feature abstraction. *IEEE Transactions on Medical Imaging*, 39(11):3619–3629, 2020.

[55] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[56] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 27503–27516, 2021.

[57] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[58] A Gammerman, V Vovk, and V Vapnik. Learning by transduction. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 148–155, 1998.

[59] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *ournal of Machine Learning Research*, 17(1):2096–2030, 2016.

[60] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, Pablo Martinez-Gonzalez, and Jose Garcia-Rodriguez. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, 70:41–65, 2018.

[61] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics*, 35(6):1–12, 2016.

[62] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[63] Germán González, George R Washko, and Raúl San José 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*, pages 215–224. Springer, 2018.

[64] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[65] Alison Gopnik, Andrew N Meltzoff, and Patricia K Kuhl. *The scientist in the crib: Minds, brains, and how children learn.* William Morrow & Co, 1999.

[66] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *IEEE International Joint Conference on Neural Networks*, volume 2, pages 729–734, 2005.

[67] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Pires, Zhaohan Guo,

Mohammad Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284, 2020.

[68] Pengfei Guo, Puyang Wang, Jinyuan Zhou, Shanshan Jiang, and Vishal M Patel. Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2423–2432, 2021.

[69] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[70] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742. IEEE, 2006.

[71] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[72] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[73] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[74] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[75] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[76] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, volume 32, pages 15663–15674, 2019.

[77] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018.

[78] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2021.

[79] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960, 2020.

[80] Haochen Hua, Yutong Li, Tonghe Wang, Nanqing Dong, Wei Li, and Junwei Cao. Edge computing with artificial intelligence: A machine learning perspective. *ACM Computing Surveys*, 2022.

[81] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[82] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. In *Advances in Neural Information Processing Systems*, volume 34, pages 677–689, 2021.

[83] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

[84] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021.

[85] Pankaj Jalote. *Fault tolerance in distributed systems*. Prentice-Hall, Inc., 1994.

[86] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

[87] Edwin T Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620, 1957.

[88] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency & disjoint learning. In *International Conference on Learning Representations*, 2021.

[89] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020.

[90] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

[91] Bhavesh D Kevadiya, Jatin Machhi, Jonathan Herskovitz, Maxim D Oleynikov, Wilson R Blomberg, Neha Bajwa, Dhruvkumar Soni, Srijanee Das, Mahmudul Hasan, Milankumar Patel, et al. Diagnostics for sars-cov-2 infections. *Nature Materials*, 20(5):593–605, 2021.

[92] Vijay Khatri and Carol V Brown. Designing data governance. *Communications of the ACM*, 53(1):148–152, 2010.

[93] Tom Kibble and Frank H Berkshire. *Classical mechanics*. World Scientific Publishing Company, 2004.

[94] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the 30th International Conference on Very Large Data Bases*, pages 180–191, 2004.

[95] Jang-Hyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with supermodular diversity. In *International Conference on Learning Representations*, 2021.

[96] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020.

[97] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[98] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[99] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning Deep Learning Workshop*, 2015.

[100] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[101] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[102] Yann Lecun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. *Predicting structured data*, 2006.

[103] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 31, pages 7167–7177, 2018.

[104] Mu Li, David G Andersen, Alexander J Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems*, pages 19–27, 2014.

[105] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[106] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[107] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

[108] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[109] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[110] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv preprint arXiv:2011.13677*, 2020.

[111] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475, 2020.

[112] Weiwei Liu, Haobo Wang, Xiaobo Shen, and Ivor W Tsang. The emerging trends of multi-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7955–7974, 2021.

[113] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[114] Angshul Majumdar. Blind denoising autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, 30(1):312–317, 2018.

[115] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[116] Alan D McNaught, Andrew Wilkinson, et al. *Compendium of chemical terminology*, volume 1669. Blackwell Science Oxford, 1997.

[117] Abid Mehmood, Iynkaran Natgunanathan, Yong Xiang, Guang Hua, and Song Guo. Protection of big data privacy. *IEEE Access*, 4:1821–1834, 2016.

[118] Charles E Metz. Basic principles of roc analysis. In *Seminars in Nuclear Medicine*, volume 8, pages 283–298. Elsevier, 1978.

[119] Ryszard S Michalski. A theory and methodology of inductive learning. In *Machine Learning*, pages 83–134. Elsevier, 1983.

[120] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[121] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[122] Tom M Mitchell. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ., 1980.

[123] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[124] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[125] Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

[126] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 2010.

[127] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, pages 1196–1204, 2013.

[128] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[129] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[130] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[131] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

[132] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[133] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[134] Olivier Petit, Nicolas Thome, Arnaud Charnoz, Alexandre Hostettler, and Luc Soler. Handling missing annotations for semantic segmentation with deep convnets. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 20–28. Springer, 2018.

[135] Jie Qin, Jiemin Fang, Qian Zhang, Wenyu Liu, Xingang Wang, and Xinggang Wang. Resizemix: Mixing data with preserved object information and true labels. *arXiv preprint arXiv:2012.11101*, 2020.

[136] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[137] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[138] Tawsifur Rahman, Amith Khandakar, Muhammad Abdul Kadir, Khandaker Rejaul Islam, Khandakar F Islam, Rashid Mazhar, Tahir Hamid, Mohammad Tariqul Islam, Saad Kashem, Zaid Bin Mahbub, et al. Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization. *IEEE Access*, 8:191586–191601, 2020.

[139] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, pages 759–766, 2007.

[140] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[141] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

[142] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A DePristo, Joshua V Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 32, pages 14707–14718, 2019.

[143] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2016.

[144] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *npj Digital Medicine*, 3(1):1–7, 2020.

[145] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, pages 1842–1850. PMLR, 2016.

[146] Karthik V Sarma, Stephanie Harmon, Thomas Sanford, Holger R Roth, Ziyue Xu, Jesse Tetreault, Daguang Xu, Mona G Flores, Alex G Raman, Rushikesh Kulkarni, et al. Federated learning improves site performance in multicenter deep learning without data sharing. *Journal of the American Medical Informatics Association*, 28(6):1259–1264, 2021.

[147] Yutaka Sasaki et al. The truth of the f-measure, 2007.

[148] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*, 2018.

[149] Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

[150] Gonglei Shi, Li Xiao, Yang Chen, and S Kevin Zhou. Marginal loss and exclusion loss for partially supervised multi-organ segmentation. *Medical Image Analysis*, page 101979, 2021.

[151] Santiago Silva, Boris A Gutman, Eduardo Romero, Paul M Thompson, Andre Altmann, and Marco Lorenzi. Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. In *IEEE International Symposium on Biomedical Imaging*, pages 270–274. IEEE, 2019.

[152] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[153] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, pages 22045–22055, 2020.

[154] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4427–4437, 2017.

[155] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 4080–4090, 2017.

[156] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951—-2959, 2012.

[157] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pages 596–608, 2020.

[158] Shiliang Sun, Honglei Shi, and Yuanbin Wu. A survey of multi-source domain adaptation. *Information Fusion*, 24:84–92, 2015.

[159] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[160] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[161] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[162] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

[163] Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 42:245–284, 2015.

[164] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[165] AFM Shahab Uddin, Mst Sirazam Monira, Wheemyung Shin, TaeChoong Chung, and Sung-Ho Bae. Saliencymix: A saliency guided data augmentation strategy for better regularization. In *International Conference on Learning Representations*, 2021.

[166] US Department of Health and Human Services. Health insurance portability and accountability act, 2017.

[167] Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *Advances in Neural Information Processing Systems*, volume 30, pages 5601–5610, 2017.

[168] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

[169] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. In *British Machine Vision Conference*, 2020.

[170] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural information Processing Systems*, pages 6000–6010, 2017.

[171] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *ACM Computing Surveys*, 53(2):1–33, 2020.

[172] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29, pages 3637–3645, 2016.

[173] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. Can multi-label classification networks know what they don't know? In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[174] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2019.

[175] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[176] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2015.

[177] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2097–2106, 2017.

[178] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, 53(3):1–34, 2020.

[179] Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, et al. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

[180] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[181] Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: an out-of-distribution detection score for variational auto-encoder. In *Advances in Neural Information Processing Systems*, volume 33, pages 20685–20696, 2020.

[182] Yanyu Xu, Xinxing Xu, Lei Jin, Shenghua Gao, Rick Siow Mong Goh, Daniel SW Ting, and Yong Liu. Partially-supervised learning for vessel segmentation in ocular images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 271–281. Springer, 2021.

[183] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021.

[184] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[185] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[186] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019.

[187] Felix Yu, Ankit Singh Rawat, Aditya Menon, and Sanjiv Kumar. Federated learning with only positive labels. In *International Conference on Machine Learning*, pages 10946–10956. PMLR, 2020.

[188] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision*, pages 6023–6032, 2019.

[189] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.

[190] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12203–12213, 2020.

[191] Guobin Zhang, Zhiyong Yang, Bin Huo, Shude Chai, and Shan Jiang. Multiorgan segmentation from partially labeled datasets with conditional nnu-net. *Computers in Biology and Medicine*, 136:104658, 2021.

[192] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[193] Jianpeng Zhang, Yutong Xie, Yong Xia, and Chunhua Shen. Dodnet: Learning to segment multi-organ and tumors from multiple partially labeled datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1195–1204, 2021.

[194] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[195] Rui Zhang, Song Guo, Junxiao Wang, Xin Xie, and Dacheng Tao. A survey on gradient inversion: Attacks, defenses and future directions. In *International Joint Conference on Artificial Intelligence*, 2022.

[196] Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 5(1):30–43, 2018.

[197] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[198] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019.

[199] Yuyin Zhou, Zhe Li, Song Bai, Chong Wang, Xinlei Chen, Mei Han, Elliot Fishman, and Alan L Yuille. Prior-aware neural network for partially-supervised multi-organ segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10672–10681, 2019.

[200] Zhi-Hua Zhou. Ensemble learning. In *Machine Learning*, pages 181–210. Springer, 2021.

[201] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.

[202] Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. Collaborative unsupervised visual representation learning from decentralized data. In *International Conference on Computer Vision*, pages 4912–4921, 2021.