Learning from Imbalanced and Incomplete Supervision with Its Application to Ride-Sharing Liability Judgment

Lan-Zhe Guo[1†], Zhi Zhou[1†], Jie-Jing Shao[1], Qi Zhang[2], Feng Kuang[2], Gao-Le Li[2], Zhang-Xun Liu[2], Guo-Bin Wu[2], Nan Ma[2], Qun Li[2], Yu-Feng Li[1§]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
{guolz,zhouz,shaojj,liyf}@lamda.nju.edu.cn
[2]Didi Chuxing, Beijing, China
{zhangqiace,kuangfeng,ligaole,liuzhangxun,wuguobin,mandymanan,liquntracy}@didiglobal.com

ABSTRACT

In multi-label tasks, sufficient and class-balanced label is usually hard to obtain, which makes it challenging to train a good classifier. In this paper, we consider the problem of learning from imbalanced and incomplete supervision, where only a small subset of labeled data is available and the label distribution is highly imbalanced. This setting is of importance and commonly appears in a variety of real applications. For instance, considering the ride-sharing liability judgment task, liability disputes usually due to a variety of reasons, however, it is expensive to manually annotate the reasons, meanwhile, the distribution of reason is often seriously imbalanced. In this paper, we present a systemic framework Limi consisting of three sub-steps, that is, *Label Separating*, *Correlation Mining* and *Label Completion*. Specifically, we propose an effective two-classifier strategy to separately tackle head and tail labels so as to alleviate the performance degradation on tail labels while maintaining high performance on head labels. Then, a novel label correlation network is adopted to explore the label relation knowledge with flexible aggregators. Moreover, the Limi framework completes the label on unlabeled instances in a semi-supervised fashion. The framework is general, flexible, and effective. Extensive experiments on diverse applications, such as the ride-sharing liability judgment task from DiDi and various benchmark tasks, demonstrate that our solution is clearly better than many competitive methods.

CCS CONCEPTS

• **Theory of computation → Semi-supervised learning**; • **Computing methodologies → Machine learning**; **Cost-sensitive learning**; **Supervised learning by classification**.

KEYWORDS

class-imbalanced learning; semi-supervised learning; multi-label learning; liability judgment

§Corresponding author
†Contribute to this work equally

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467305

ACM Reference Format:
Lan-Zhe Guo, Zhi-Zhou, Jie-Jing Shao, Qi Zhang, Feng Kuang, Gao-Le Li, Zhang-Xun Liu, Guo-Bin Wu, Nan Ma, Qun Li, Yu-Feng Li. 2021. Learning from Imbalanced and Incomplete Supervision with Its Application to Ride-Sharing Liability Judgment. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447548.3467305

1 INTRODUCTION

Learning from multi-label data (multi-label learning) [33], where each training instance is associated with multiple labels, has achieved great success in many real-world applications. These successful techniques typically require training data with sufficient and class-balanced supervised information. However, it is often the case that such strong supervision is hard to obtain due to the expensive cost of the labeling process. Therefore, it is desired to facilitate the learning system with the capability of multi-label learning from weak supervision.

We consider the problem of learning from *imbalanced* and *incomplete* supervision. Specifically, only a small subset of training data is observed with labels while the others remain unlabeled. Meanwhile, the given labels might be class-imbalanced. This setting is crucial since it commonly occurs in many real-world applications. For example, in DiDi ride-sharing liability judgment task [9], when liability disputes occur, the platform needs to decide whether the driver is responsible and predict all related decidendi reasons to make the decision convincing. As there are a large number of disputes that occur, it is not possible to label all data, and meanwhile, there is a severe class-imbalance problem since some decidendi reasons are more often encountered than others. Similar situations also occur in the image classification task, where the frequency distribution of visual categories in our daily life is inherently long-tailed [18, 26] and we usually lack the resources to create a sufficiently large images dataset [11, 19]. We illustrate the problem in Figure 1.

These two issues have been studied separately in the area of *Class-Imbalanced Multi-Label Learning* (CIMLL) [21, 26, 32] and *Semi-Supervised Multi-Label Learning* (SSMLL) [6, 23, 24]. For imbalanced supervision, CIMLL approaches manage to re-weight the loss function or re-sample the classes in order to resist the imbalance problem. However, they need sufficient labeled data and cannot utilize numerous unlabeled data. For incomplete supervision, SSMLL approaches leverage unlabeled data and limited labeled data to construct the multi-label predictor. However, when the label

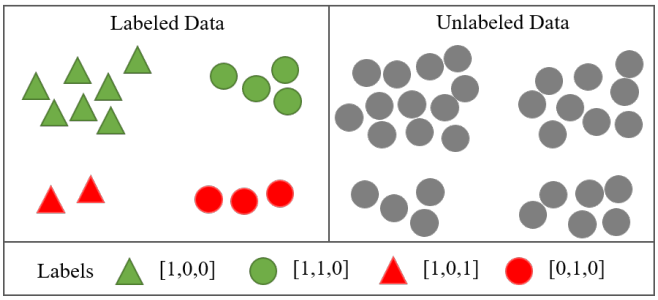

Figure 1: A simple 2-dimensional multi-label dataset with a small number of labeled data (left) and a large number of unlabeled data (right). The dataset concerns points in a plane characterized by three labels, namely the shape of the points (triangles, circles) and the color of the points (green, red). At the bottom, we see the four different label combinations that exist in the dataset. The dataset is also class imbalanced because the numbers of relevant instances for the three labels are 13, 7, 2 respectively.

distribution is imbalanced, these approaches suffer severe performance degradation problems. Therefore, it is much appreciated for approaches that are able to deal with the class-imbalanced and unlabeled data simultaneously.

There lack of relevant studies for the problem of learning with imbalanced and incomplete supervision simultaneously, where there are a vast number of unlabeled data and a limited number of class-imbalanced labeled data. This problem turns out quite challenging and it is not trivial to combines the advantages of CIMLL and SSMLL approaches to address this problem. For traditional CIMLL approaches, on one hand, labeled data are insufficient to estimate the underlying class-imbalance ratio which is essential for these approaches. On the other hand, these approaches are not able to access label information from unlabeled data, and thus cannot leverage the incomplete supervision to alleviate the class imbalance. For traditional SSMLL approaches, to handle a vast number of unlabeled data, an underlying assumption is that the label distribution should be balanced. Otherwise, the imbalanced label distribution can significantly mislead the learning system. For instance, we present the results of state-of-the-art CIMLL approach *DBL* [26], SSMLL approach *DRML* [23] and a simple combination of these two approaches *DRML+DBL* in Figure 2. We can see that all these methods suffer from performance degradation issues compared to the simple baseline method that directly trains a fully connected neural network by minimizing the binary cross-entropy (BCE) loss.

To address this challenging yet realistic problem, we propose a novel framework Limi (Learning IMbalanced and Incomplete supervision) in this paper. Limi is consisted of three sub-modules: *label separating*, *correlation mining* and *label completion*. Specifically, we deploy two models to deal with head labels (frequently occurring labels) and tail labels (infrequently occurring labels) separately in order to alleviate the performance degradation on tail labels while maintaining high performance on head labels. Then, after obtaining the separate prediction results, a novel label correlation network is proposed to explore the label correlation knowledge which is of paramount importance for multi-label data. Meanwhile, the framework can incorporate different semi-supervised assumptions flexibly to exploit the unlabeled instances. Extensive experimental results on diverse real applications, such as the ride-sharing liability judgment tasks from DiDi and various benchmark tasks, clearly demonstrate the effectiveness of our framework.

We summarize our main contributions as following:

(1) We study the problem of learning from imbalanced and incomplete supervision simultaneously, which occurs in many real-world applications but has rarely studied.
(2) We propose a novel systemic multi-label learning framework Limi, which is able to address the class imbalance effectively, explore label correlation generically and utilize unlabeled instances flexibly.
(3) We conduct extensive empirical studies on both real-world applications and benchmark datasets to show the superiority of our proposed framework.

In the following, we first review several related works in section 2, then present the technical details of the proposed Limi framework in section 3. Next, we report the empirical results in section 4. Finally, we conclude this paper.

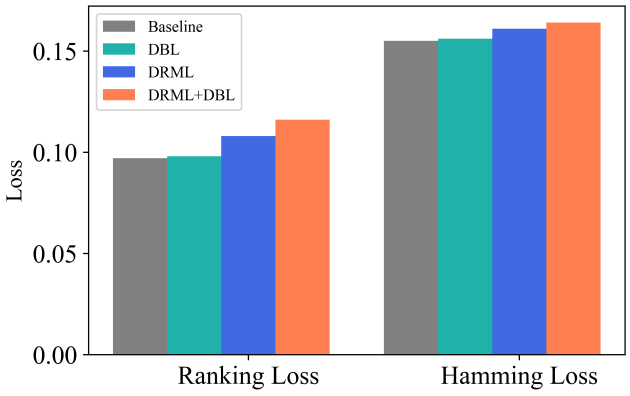

Figure 2: Experiments on DiDi liability judgment data set with 2,000 labeled data and 18,000 unlabeled data. The data set is highly class imbalanced. Two representative multi-label metrics hamming loss and ranking loss are reported.

2 RELATED WORK

Our paper is related to two branches of studies, that is, class-imbalanced multi-label learning and semi-supervised multi-label learning.

Class-imbalanced learning solves the problem where the training data has a high skewed label distribution [15]. Existing methods for dealing with the class imbalance problem in multi-label data can be separated into two lines: multi-label re-sampling and label re-weighting. The first line tries to reduce the imbalance level of multi-label data via under-sampling or over-sampling techniques as a pre-processing step. For example, MLRUS [3] omits instances with head labels randomly to alleviate the imbalance in each individual label. As a twin method of MLRUS, MLROS [3] increases the frequency of tail labels by replicating instances relevant to tail labels

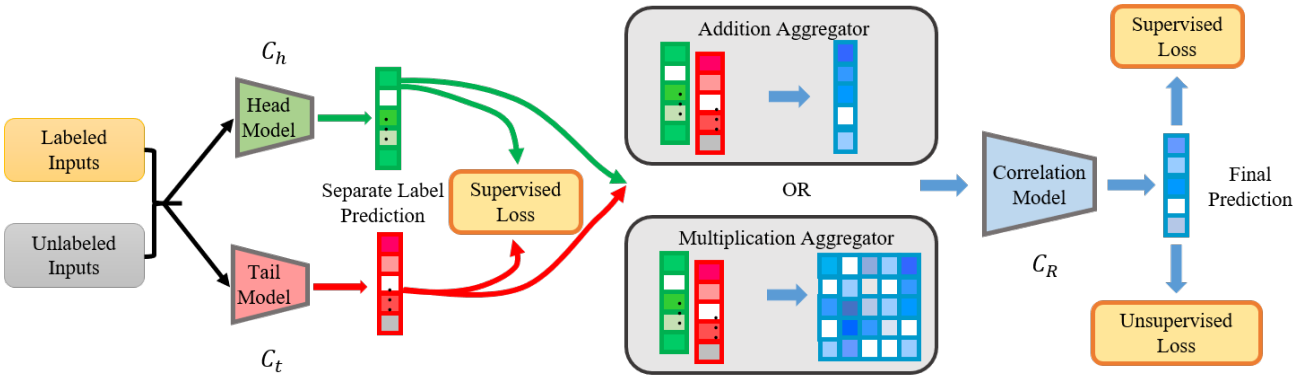

Figure 3: The proposed Limi framework. $C_h(\cdot)$ and $C_t(\cdot)$ are two models to address head and tail labels separately. The two separate predictions from $C_h(\cdot)$ and $C_t(\cdot)$ are forwarded to the label correlation network $C_R(\cdot)$ to further explore the label relations and we provide two aggregation choices for label correlation mining. After getting the final prediction, unsupervised regularization based on different semi-supervised assumptions is adopted to exploit unlabeled instances.

in each label. MLeNN [2] is an undersampling algorithm based on the ENN (Edited Nearest Neighbor) rule. To reduce the risk of overfitting caused by replicating instances, MLSMOTE [4] generates the classical SMOTE algorithm to multi-label data by randomly select instances containing tail labels, along with its neighbors, to generate synthetic instances. The second line focuses on the multi-label learning algorithm handling different classes by different weights. For example, COCOA [32] converts the original multi-label data to several multi-class datasets for each label and builds imbalance classifiers with the assistance of weighting for each dataset. SOSHF [5] transforms the multi-label data to an imbalanced single label classification assignment via cost-sensitive clustering and the new task is addressed by oblique structured Hellinger decision trees. [26] proposed a modified distribution balance BCE loss that takes the label co-occurrence into consideration and achieved SOTA performance in long-tailed multi-label datasets. However, these methods need a large number of labeled data to estimate the class imbalance ratio and could not perform well in the semi-supervised scenario.

Semi-supervised learning aims to improve the learning performance by utilizing labeled as well as unlabeled data simultaneously. [17] formulated the semi-supervised multi-label learning problem as a constrained non-negative matrix factorization problem by assuming that similar instances should have similar predicted labels. [25] took advantage of label correlation in labeled instances and of maximum-margin regularization over unlabeled instances to optimize linear predictors. [22] introduced the SMILE method which uses a graph to embody both labeled and unlabeled instances and trains a graph-regularized semi-supervised linear classifier. [28, 31] adapted the co-training approach to multi-label data which optimize two disjoint feature views by maximizing the diversity and iteratively communicates the pairwise ranking predictions of either classifier o unlabeled instances. [23] introduced a DRML approach that jointly explores the feature distribution and the label relation simultaneously by adopting a domain adaptation strategy and generating pseudo labels for unlabeled data. However, these methods ignore the inherent class imbalance problem and could suffer severe performance degradation problem.

Table 1: Summary of Notations.

Notation	Meaning
n	Number of labeled instances
m	Number of unlabeled instances
k	Number of labels
$\mathbf{x} \in \mathcal{X} = \mathbb{R}^d$	Feature vector of instances
$\mathbf{y} \in \mathcal{Y} = \{0,1\}^k$	Ground-truth label of instances
$\hat{\mathbf{y}} \in \mathcal{Y}$	Predicted label of instances
R_i	Aggregated result of $C_h(\mathbf{x}_i)$ and $C_t(\mathbf{x}_i)$
$f : \mathcal{X} \to \mathcal{Y}$	Learned function
$C_h(\mathbf{x}) \in [0,1]^k$	Predicted probabilities of head model
$C_t(\mathbf{x}) \in [0,1]^k$	Predicted probabilities of tail model
$f(\mathbf{x}) \in [0,1]^k$	Predicted probabilities of Limi framework

3 LEARNING FROM IMBALANCED AND INCOMPLETE SUPERVISION

In this section, we propose a systemic framework Limi to deal with multi-label data with imbalanced and incomplete supervision. Three major challenges to the class-imbalanced semi-supervised multi-label data are i) The tail labels with insufficient instances are difficult to learn reliably. How to avoid performance degradation on tail labels? ii) The label correlation is essential for the performance of multi-label learning algorithms. How to explore the label correlation knowledge effectively? iii) The labeled instances are insufficient to learn a good model. How to make use of unlabeled data to improve performance? Limi provides a systemic solution consisting three main modules, *label separating*, *correlation mining* and *label completion*. We first present some notations used in the following and then describe the detail of the main techniques.

3.1 Preliminaries

Let $\mathcal{X} = \mathbb{R}^d$ be a d-dimensional input feature space and $\mathcal{Y} = \{0,1\}^k$ a k-dimensional label space. Given $\mathcal{D}_l = \{\mathbf{X}_l, \mathbf{Y}_l\} = \{(\mathbf{x}_i, \mathbf{y}_i) | 1 \le i \le n\}$ be the labeled training data containing n instances. $\mathbf{X}_l \in$

$\mathbb{R}^{n\times d}$ is the feature matrix and $\mathbf{Y}_l \in \{0,1\}^{n\times k}$ is the label matrix. Each instance $(\mathbf{x}_i, \mathbf{y}_i)$ consists of a feature vector $\mathbf{x}_i \in \mathcal{X}$ and a label vector $\mathbf{y}_i \in \mathcal{Y}$. $y_i^j = 1(0)$ indicates that j-th label is (not) relevant with the i-th instance. Meanwhile, we have unlabeled dataset $\mathcal{D}_u = \{\mathbf{X}_u\} = \{\mathbf{x}_i | n+1 \le i \le n+m\}$ that contains m unlabeled instances and $\mathbf{X}_u \in \mathbb{R}^{m\times d}$. The goal is to learn a mapping function $f : \mathcal{X} \to \mathcal{Y}$ such that for an unseen instance $\mathbf{x} \in \mathcal{X}$, one predicts its label vector $\hat{\mathbf{y}} \in \mathcal{Y}$ correctly.

3.2 Label Separating

For the class-imbalanced problem, simply train a model on all data is sub-optimal as the model under-fits for tail labels, leading to low performance. Re-sampling or re-weighting on the whole dataset can alleviate the performance degradation to a certain extent. However, these methods need a large number of labeled instances to estimate the class imbalance ratio and therefore could not perform well with incomplete supervision. To alleviate the discrepancy between head and tail labels, we design a two-classifier strategy that treats head and tail labels with different techniques for the reason that it is difficult to maintain satisfying performance on all labels with a single model when the labeled data is scarce.

Specifically, for head labels, we can simply train a neural network $C_h(\cdot)$ by minimizing the standard BCE loss, for the reason that the model trained with original imbalanced label distribution will naturally lead to good performance for head labels. The objective can be written as:

$$\min_{C_h} \ell_h(C_h(\mathbf{X}_l), \mathbf{Y}_l) = BCE(C_h(\mathbf{X}_l), \mathbf{Y}_l) \tag{1}$$

ℓ_h indicates the loss function for head model and

$$BCE(C_h(\mathbf{x}_i), \mathbf{y}_i) = \frac{1}{k}\sum_{j=1}^{k}[y_i^j \log(C_h(\mathbf{x}_i)_j) + (1-y_i^j)\log(1-C_h(\mathbf{x}_i)_j)] \tag{2}$$

where $C_h(\mathbf{x}_i) \in [0,1]^k$ is the predicted label vector for instance $\mathbf{x}$ of head model, and $C_h(\mathbf{x}_i)_j$ indicates the predicted probability that whether $\mathbf{x}_i$ is relevant with label j, y_i^j is the true label of instance $\mathbf{x}_i$ on label j.

For tail labels, we propose to re-weight the loss function to improve their performance. It is noteworthy that it is much easier to achieve good performance only on tail labels by re-weighting method compared with performing well on all labels simultaneously. Without taking label co-occurrence into consideration, for each instance i and class j with $y_i^j = 1$, the expectation of *class-level* sampling frequency can be calculated as:

$$P_j^C(x_i) = \frac{1}{k}\frac{1}{n_j} \tag{3}$$

where n_j denote the number of training instances that relevant with label j.

However, in the multi-label scenario, simply re-weighting strategies based on the label frequency could not work well because an instance usually contains several ground-truth labels and makes the re-weighting strategy for labels no longer independent. Therefore, in addition to the class-level sampling frequency, we also consider the *instance-level* sampling frequency. For an instance x_i and it corresponding label vector y_i, it is supposed to be repeatedly sampled by each positive label j it contains, thus the expectation of instance-level sampling frequency can be estimated as:

$$P^I(x_i) = \frac{1}{k}\sum_{y_i^j=1}\frac{1}{n_j} \tag{4}$$

Correspondingly, we can calculate a re-balancing weight r_i^j to close the gap between expected sampling times and actual sampling times:

$$\mathbf{r}_i^j = \frac{P_j^C(\mathbf{x}_i)}{P^I(\mathbf{x}_i)} \tag{5}$$

Therefore, the tail model can be trained by minimizing the following re-weighted BCE loss:

$$\ell_t(C_t(\mathbf{X}_l), Y_l) = \frac{1}{nk}\sum_{i=1}^{n}\sum_{j=1}^{k} BCE(\mathbf{x}_i, y_i^j)\cdot r_i^j \tag{6}$$

In this way, the separate predicted results of the head-model $C_h(\cdot)$ and the tail-model $C_t(\cdot)$ could be obtained by optimizing the following objective:

$$\min_{C_h, C_t} \ell_h(C_h(X_l), Y_l) + \ell_t(C_t(X_l), Y_l) \tag{7}$$

3.3 Correlation Mining

Averaging these two outputs from classifier $C_h(\cdot)$ and $C_t(\cdot)$ is a straightforward to obtain the final prediction. However, it is well-known that label correlation is crucial to further improve the learning performance for multi-label problems [33]. To this end, we further propose a novel and effective label-level correlation network, $C_R(\cdot)$ with two flexible aggregators to automatically explore the label correlation knowledge:

Addition Aggregator. Our first candidate aggregator function is the addition aggregator, where we take a weighted sum of the two prediction results $C_h(\mathbf{x})$ and $C_t(\mathbf{x})$ and obtain

$$R_i = \mathbf{w}_h C_h(\mathbf{x}_i) + \mathbf{w}_t C_t(\mathbf{x}_i) \tag{8}$$

The obtained $R_i \in \mathbb{R}^{1\times k}$ is then transformed into a new label space to obtain the final prediction result $C_R(w_h C_h(\mathbf{x}) + w_t C_t(\mathbf{x}))$. And the weight w and parameter of $C_R(\cdot)$ can be optimized simultaneously.

The label correlation network $C_R(\cdot)$ with the addition aggregator can be regarded as a label projection that projects the label vector into a new label space and allows us to compute label difference in the new space.

Multiplication Aggregator. We also examine a more complex aggregator by multiplying the transposition of $C_h(\mathbf{x})$ and $\mathbf{C}_t(\mathbf{x})$ and obtain

$$R_i = C_h(\mathbf{x}_i)^\top \times C_t(\mathbf{x}_i) \tag{9}$$

where $R_i \in \mathbb{R}^{k\times k}$ is the correlation matrix.

The obtained R_i is reshaped to a $\mathbb{R}^{1\times k^2}$ vector and forwarded to a fully connected relation network $C_R(\cdot)$. $C_R(\cdot)$ further returns the final multi-label prediction result based on R_i.

The multiplication aggregator can be considered as a dot-product similarity metric of the pairwise labels [23]. Thus, $C_R(\cdot)$ explores the latent correlation knowledge residing inside the training data based on the obtained similarities and further refine the predicted results from $C_h(\cdot)$ and $C_t(\cdot)$ to improve the performance.

After obtaining the predicted result, the objective of the correlation network can be written as:

$$\min_{C_R} \ell_R(C_R(\mathbf{X}_l, \mathbf{Y}_l)) = BCE(C_R(R_i), \mathbf{Y}_l) \tag{10}$$

In the training procedure, $C_R(\cdot)$ is trained simultaneously with $C_h(\cdot)$ and $C_t(\cdot)$, i.e.,

$$\min_{C_R, C_h, C_t} \ell_h + \ell_t + \ell_R \tag{11}$$

Empirical evidences show that for extreme multi-label data (i.e., multi-label classification with many labels), the addition aggregator can achieve better performance while multiplication aggregator can explore the label correlation more effectively when the label dimension is not very high.

3.4 Label Completion

The above framework can be easily deployed for labeled samples. Meanwhile, how to exploit a large number of unlabeled instances is also an essential part in our setting. To utilize the information behind unlabeled data safely, we should adopt suitable assumptions for different data set structures, otherwise, semi-supervised algorithms could suffer performance degradation problem [16]. We consider two commonly used semi-supervised regularizations based on different assumptions:

Consistency Regularization: The most commonly used regularization for unlabeled instances is the consistency regularization, which is derived from the manifold assumption that two instances tend to have a large overlap in their assigned label memberships if they share high similarity in their input patterns [17].

Specifically, let the instance similarity matrix be $\mathbf{S}$ that can be defined with RBF kernel, i.e., $S_{ij} = \exp\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}$. Denote the final predicted probability of our framework for instance $\mathbf{x}$ is $f(\mathbf{x})$, i.e., $f(\mathbf{x}) = C_R(C_h(\mathbf{x}), C_t(\mathbf{x}))$, the unsupervised regularization term can be written as:

$$\begin{aligned} \ell_u(\mathbf{X}) &= \frac{1}{2}\sum_{i=1}^{n+m}\sum_{j=1}^{n+m}(\mathbf{S}_{ij})(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 \\ &= f(\mathbf{X})^\top \mathbf{L} f(\mathbf{X}) \end{aligned} \tag{12}$$

where $\mathbf{X} = [\mathbf{X}_l, \mathbf{X}_u]$, $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is the Laplacian matrix of $\mathbf{S}$ and $\mathbf{D}$ is a diagonal matrix with elements $D_{ii} = \sum_{j=1}^{n+m} S_{ij}, i = 1, \cdots, n+m$.

Large Margin Principle: According to the *No Free Lunch Theorem*, we know that there is no algorithm suitable for all data sets. For some data set that does not satisfy manifold assumption, large margin principle is another common choice. Large margin principle is based on the underlying assumption that the classifier's decision boundary should not pass through high-density regions of the marginal data distribution [1, 10].

Specifically, we adopt the hinge loss function $H(t) = \max(0, 1-t)$ to evaluate the loss on unlabeled instances. Different from binary classification task, we need to compute the hinge loss on every single label as following:

$$\ell_u(\mathbf{X}) = \sum_{i=1}^{n+m}\sum_{j=1}^{k} H(2 * |f(\mathbf{x}_i)_j - 0.5|) \tag{13}$$

Table 2: Definition of 9 Multi-Label Performance Metrics

Measure	Formulation
Hamming loss	$\frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K}\mathbb{I}(\hat{y}_{ij} \neq y_{ij})$
Ranking Loss	$\frac{1}{N}\sum_{i=1}^{N}\frac{\lvert\mathcal{S}^i_{rank}\rvert}{\lvert Y^+_{i\cdot}\rvert\lvert Y^-_{i\cdot}\rvert}$
One Error	$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\text{argmax}\, f(x_i) \notin Y^+_{i\cdot})$
Coverage	$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\max_{j\in Y^+_{i\cdot}} rank_f(\mathbf{x}_i, j) - 1)$
Average Precision	$\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\lvert Y^+_{i\cdot}\rvert}\sum_{j\in Y^+_{i\cdot}}\frac{\lvert\mathcal{S}^{ij}_{precision}\rvert}{rank_f(\mathbf{x}_i, j)}$
Macro AUC	$\frac{1}{K}\sum_{j=1}^{K}\frac{\lvert\mathcal{S}^j_{macro}\rvert}{\lvert Y^+_{\cdot j}\rvert\lvert Y^-_{\cdot j}\rvert}$
Micro AUC	$\frac{\lvert\mathcal{S}_{micro}\rvert}{(\sum_{i=1}^{N}\lvert Y^+_{i\cdot}\rvert)\cdot(\sum_{i=1}^{N}\lvert Y^-_{i\cdot}\rvert)}$
Macro F1	$\frac{1}{K}\sum_{j=1}^{K}\frac{2\sum_{i=1}^{N} y_{ij}\hat{y}_{ij}}{\sum_{i=1}^{N} y_{ij} + \sum_{i=1}^{N}\hat{y}_{ij}}$
Micro F1	$\frac{2\sum_{j=1}^{K}\sum_{i=1}^{N} y_{ij}\hat{y}_{ij}}{\sum_{j=1}^{K}\sum_{i=1}^{N} y_{ij} + \sum_{j=1}^{K}\sum_{i=1}^{N}\hat{y}_{ij}}$

$$\mathcal{S}^i_{rank} = \{(u,v) | f(\mathbf{x}_i)_u \leq f(\mathbf{x}_i)_v, (u,v) \in Y^+_{i\cdot} \times Y^-_{i\cdot}\}$$
$$\mathcal{S}^{ij}_{precision} = \{k \in Y^+_{i\cdot} | rank_f(\mathbf{x}_i, k) \leq rank_f(\mathbf{x}_i, j)\}$$
$$\mathcal{S}^j_{macro} = \{(a,b) \in Y^+_{\cdot j} \times Y^-_{\cdot j} | f(x_a)_j \geq f(x_b)_j\}$$
$$\mathcal{S}_{micro} = \{(a,b,i,j) | (a,b) \in Y^+_{\cdot i} \times Y^-_{\cdot j}, f(\mathbf{x}_a)_i \geq f(\mathbf{x}_b)_j\}$$

By optimizing the above unsupervised regularization term, the decision boundary is pushed to less dense areas and result in a large margin classifier.

Overall, our framework Limi contains three modules $C_h(\cdot), C_t(\cdot)$ and $C_R(\cdot)$, which are jointly optimized by minimizing the following objective:

$$\min_{C_h, C_t, C_R} \ell_h + \ell_t + \ell_R + \lambda\ell_u \tag{14}$$

where λ is a trade-off hyper-parameter that balances the contribution of supervised and unsupervised loss functions. The overall framework is illustrated in Figure 2.

4 EXPERIMENTS

In this section, we conduct extensive experiments on a real-word industrial applications and various benchmark tests to evaluate the effectiveness of the proposal Limi approach.

4.1 Experimental Setup

Compared Methods. The proposed approach is compared with a number of methods. First, we conduct a baseline FCN method, which directly train a Fully Connected Network by minimizing the standard BCE loss. Then, we evaluate a representative SOTA multi-label learning method CAMEL [8], which is a novel multi-label learning approach that first learn the label correlations via sparse reconstruction in the label space, and then integrate the learned label correlations into model training. We also compare with two SOTA class-imbalanced multi-label learning methods: a) DBL [26], which tries to re-balance the weight of each label and takes into account the impact caused by label co-occurrence in multi-label data; b) DBL+NT [26], which further proposed a negative tolerant

Table 3: Experimental results (mean±std) on *DiDi Liability Judgment* dataset. ↑ (↓) indicates the larger (smaller) the better. The best performance and its comparable performances are bolded.

5% Labeled Instances									
Methods	Hamming Loss ↓	Ranking Loss ↓	One Error ↓	Coverage ↓	Ave. Precision ↑	Macro AUC ↑	Micro AUC ↑	Macro F1 ↑	Micro F1 ↑
FCN	0.160±0.008	0.100±0.001	0.237±0.001	0.121±0.001	0.839±0.001	0.588±0.012	0.885±0.004	0.239±0.014	0.584±0.016
CAMEL	0.159±0.006	0.115±0.003	0.243±0.002	0.135±0.002	0.831±0.001	**0.593±0.006**	0.878±0.002	0.239±0.006	0.585±0.017
DBL	0.160±0.008	0.100±0.001	**0.236±0.001**	0.121±0.001	**0.840±0.001**	0.586±0.017	0.885±0.005	0.237±0.017	0.583±0.015
DBL+NT	0.157±0.006	0.100±0.001	0.237±0.002	0.121±0.001	0.839±0.001	0.592±0.009	0.884±0.003	**0.243±0.007**	0.591±0.013
PL	0.160±0.008	0.100±0.001	0.237±0.002	0.121±0.001	0.839±0.001	0.586±0.012	0.886±0.003	0.237±0.010	0.584±0.016
DRML	0.161±0.013	0.108±0.002	0.264±0.003	0.127±0.002	0.825±0.002	0.524±0.016	0.870±0.005	0.217±0.007	0.579±0.026
DRML+DBL	0.164±0.009	0.116±0.008	0.276±0.019	0.135±0.006	0.816±0.011	0.533±0.011	0.858±0.010	0.221±0.005	0.574±0.022
Proposal	**0.152±0.010**	**0.099±0.001**	**0.236±0.002**	**0.120±0.001**	**0.840±0.001**	0.589±0.025	**0.889±0.004**	0.232±0.014	**0.597±0.019**
10% Labeled Instances									
Methods	Hamming Loss ↓	Ranking Loss ↓	One Error ↓	Coverage ↓	Ave. Precision ↑	Macro AUC ↑	Micro AUC ↑	Macro F1 ↑	Micro F1 ↑
FCN	0.155± 0.006	0.098±0.001	0.237±0.002	0.120±0.001	0.841±0.001	0.607±0.016	0.889±0.004	0.249±0.015	0.596±0.013
CAMEL	0.156±0.003	0.112±0.001	0.240±0.002	0.132±0.001	0.834±0.001	0.606±0.005	0.882±0.001	0.248±0.007	0.593±0.009
DBL	0.156±0.006	0.098±0.002	0.236±0.001	0.120±0.001	0.841±0.001	0.604±0.019	0.889±0.004	0.249±0.013	0.596±0.014
DBL+NT	0.154±0.006	0.098±0.001	0.236±0.001	0.120±0.001	0.841±0.001	0.601±0.018	0.889±0.004	0.245±0.013	0.599±0.012
PL	0.154±0.005	0.098±0.001	0.237±0.002	0.120±0.001	0.840±0.001	0.613±0.007	0.891±0.002	0.255±0.007	**0.600±0.009**
DRML	0.160±0.006	0.103±0.002	0.255±0.011	0.123±0.001	0.832±0.005	0.530±0.011	0.876±0.004	0.223±0.005	0.584±0.012
DRML+DBL	0.160±0.008	0.105±0.002	0.252±0.011	0.125±0.002	0.831±0.005	0.533±0.013	0.871±0.004	0.222±0.005	0.585±0.017
Proposal	**0.150±0.005**	**0.096±0.002**	**0.234±0.001**	**0.115±0.002**	**0.843±0.002**	**0.614±0.023**	**0.892±0.010**	**0.260±0.010**	0.599±0.012

Table 4: Experimental results (mean±std) on *CUB* dataset. ↑ (↓) indicates the larger (smaller) the better. The best performance and its comparable performances are bolded.

5% Labeled Instances									
Methods	Hamming Loss ↓	Ranking Loss ↓	One Error ↓	Coverage ↓	Ave. Precision ↑	Macro AUC ↑	Micro AUC ↑	Macro F1 ↑	Micro F1 ↑
FCN	0.123±0.004	0.142±0.002	0.191±0.016	0.597±0.007	0.503±0.005	0.679±0.005	0.852±0.002	0.225±0.007	0.413±0.011
CAMEL	**0.114±0.001**	0.158±0.001	0.186±0.011	0.786±0.007	0.496±0.003	0.682±0.006	0.842±0.001	**0.245±0.003**	0.430±0.006
DBL	0.121±0.004	0.140±0.003	0.178±0.021	0.587±0.007	0.504±0.006	0.684±0.007	0.856±0.003	0.230±0.011	0.414±0.013
DBL+NT	0.123±0.003	0.142±0.003	**0.150±0.008**	0.589±0.007	0.503±0.006	0.680±0.008	0.854±0.003	0.225±0.009	0.410±0.010
PL	0.118±0.002	0.138±0.002	0.201±0.014	0.590±0.007	0.508±0.005	0.683±0.004	0.852±0.003	0.239±0.003	0.424±0.008
DRML	0.121±0.004	0.166±0.002	0.265±0.034	0.706±0.013	0.464±0.007	0.621±0.003	0.831±0.003	0.193±0.005	0.395±0.008
DRML+DBL	0.121±0.002	0.156±0.002	0.161±0.017	0.625±0.008	0.484±0.003	0.645±0.006	0.841±0.002	0.200±0.004	0.399±0.005
Proposal	**0.114±0.002**	**0.134±0.002**	0.165±0.018	**0.585±0.007**	**0.512±0.007**	**0.688±0.004**	**0.858±0.004**	0.240±0.004	**0.436±0.009**
10% Labeled Instances									
Methods	Hamming Loss ↓	Ranking Loss ↓	One Error ↓	Coverage ↓	Ave. Precision ↑	Macro AUC ↑	Micro AUC ↑	Macro F1 ↑	Micro F1 ↑
FCN	0.115±0.002	0.131±0.003	0.183±0.013	0.562±0.007	0.525±0.007	0.708±0.007	0.865±0.002	0.254±0.009	0.440±0.007
CAMEL	0.112±0.001	0.151±0.002	0.174±0.004	0.767±0.005	0.510±0.003	0.691±0.005	0.849±0.002	0.256±0.005	0.444±0.004
DBL	0.115±0.002	0.130±0.002	0.160±0.013	0.552±0.006	0.525±0.005	0.707±0.006	0.867±0.003	0.255±0.005	0.438±0.004
DBL+NT	0.117±0.002	0.132±0.002	**0.144±0.010**	0.558±0.006	0.523±0.006	0.703±0.005	0.866±0.002	0.249±0.006	0.431±0.004
PL	0.114±0.002	**0.129±0.002**	0.178±0.014	0.556±0.006	0.526±0.005	**0.710±0.006**	0.865±0.002	**0.258±0.004**	0.441±0.003
DRML	0.116±0.001	0.157±0.004	0.277±0.032	0.676±0.014	0.475±0.006	0.641±0.008	0.840±0.003	0.214±0.009	0.418±0.003
DRML+DBL	0.117±0.002	0.148±0.003	0.185±0.033	0.609±0.007	0.493±0.009	0.670±0.010	0.849±0.003	0.221±0.009	0.420±0.004
Proposal	**0.102±0.002**	**0.129±0.002**	0.151±0.016	**0.551±0.008**	**0.527±0.006**	**0.710±0.004**	**0.869±0.002**	0.255±0.006	**0.445±0.003**

regularization along with the DBL to mitigate the over-suppression of negative labels. For semi-supervised learning, we compare with two methods: a) PL [14], which is a simple and efficient method by assigning high confidence pseudo-labels to unlabeled instances

Table 5: Experimental results (mean±std) on *BibTex* dataset. ↑ (↓) indicates the larger (smaller) the better. The best performance and its comparable performances are bolded.

5% Labeled Instances									
Methods	Hamming Loss ↓	Ranking Loss ↓	One Error ↓	Coverage ↓	Ave. Precision ↑	Macro AUC ↑	Micro AUC ↑	Macro F1 ↑	Micro F1 ↑
FCN	0.025±0.001	0.163±0.006	0.572±0.012	0.263±0.008	0.379±0.009	0.686±0.016	0.727±0.015	0.113±0.013	0.203±0.017
CAMEL	**0.024±0.001**	0.176±0.005	0.587±0.015	0.290±0.008	0.366±0.008	**0.803±0.006**	**0.825±0.005**	**0.210±0.008**	**0.284±0.009**
DBL	0.025±0.001	0.163±0.006	0.568±0.011	0.264±0.008	0.382±0.009	0.680±0.016	0.723±0.016	0.115±0.013	0.204±0.016
DBL+NT	**0.024±0.001**	0.164±0.005	**0.563±0.013**	0.267±0.008	0.386±0.008	0.693±0.017	0.735±0.016	0.133±0.013	0.224±0.017
PL	0.027±0.001	0.182±0.006	0.594±0.021	0.291±0.008	0.354±0.013	0.594±0.033	0.635±0.033	0.079±0.013	0.153±0.017
DRML	0.024±0.002	0.343±0.009	0.802±0.042	0.490±0.012	0.161±0.029	0.571±0.035	0.618±0.018	0.060±0.019	0.144±0.033
DRML+DBL	0.028±0.002	0.289±0.011	0.747±0.021	0.426±0.012	0.211±0.016	0.585±0.014	0.635±0.011	0.044±0.008	0.102±0.015
Proposal	**0.024±0.001**	**0.160±0.007**	**0.563±0.014**	**0.262±0.009**	**0.387±0.011**	0.696±0.013	0.737±0.013	0.130±0.013	0.206±0.014
10% Labeled Instances									
Methods	Hamming Loss ↓	Ranking Loss ↓	One Error ↓	Coverage ↓	Ave. Precision ↑	Macro AUC ↑	Micro AUC ↑	Macro F1 ↑	Micro F1 ↑
FCN	**0.021±0.001**	0.127±0.003	0.508±0.008	0.210±0.004	0.438±0.006	0.755±0.009	0.795±0.008	0.156±0.009	0.264±0.014
CAMEL	**0.021±0.001**	0.144±0.003	0.508±0.006	0.248±0.007	0.436±0.004	**0.833±0.006**	**0.856±0.005**	**0.256±0.008**	**0.334±0.008**
DBL	**0.021±0.001**	0.127±0.003	0.508±0.009	0.210±0.004	0.440±0.006	0.751±0.009	0.791±0.008	0.155±0.010	0.263±0.015
DBL+NT	**0.021±0.001**	0.128±0.003	0.507±0.008	0.214±0.005	0.444±0.006	0.757±0.010	0.797±0.009	0.173±0.010	0.280±0.014
PL	0.022±0.001	0.133±0.004	0.519±0.008	0.218±0.008	0.428±0.006	0.705±0.017	0.747±0.016	0.125±0.014	0.228±0.018
DRML	0.022±0.004	0.310±0.008	0.731±0.018	0.450±0.010	0.214±0.010	0.658±0.021	0.676±0.010	0.105±0.012	0.219±0.013
DRML+DBL	0.024±0.002	0.237±0.011	0.671±0.017	0.352±0.013	0.272±0.013	0.665±0.023	0.713±0.015	0.088±0.016	0.181±0.027
Proposal	**0.021±0.001**	**0.124±0.002**	**0.506±0.006**	**0.206±0.204**	**0.446±0.006**	0.785±0.010	0.821± 0.010	0.178±0.008	0.277±0.011

based on the predicted probabilities; b) DRML [23]: which adopts two dual classifiers to align feature distributions and an label relation network to explore the label relations. Moreover, we try to improve the DRML method by replacing the standard BCE loss of DRML with the distribution balance loss, which we denote as DRML+DBL.

Performance Metrics. We measure the classification results in terms of various multi-label evaluation criteria that are both instance-wise and label-wise effective [27], including Hamming Loss, Ranking Loss, One Error, Coverage, Average Precision, Macro AUC, Micro AUC, Macro F1, Micro F1. The formulation of these metrics is shown in Table 2. For more details about the evaluation metrics please refer to [33].

Implementation Details. For each dataset, we randomly split the train, validation, and test set based on the ratio 7:1:2. We consider the ratio of labeled data by randomly select 5% and 10% training instances and the rest are unlabeled data. We train our framework for a maximum of 500 epochs using Adam with a learning rate of 0.001 and early stopping with a window size of 30. The parameter λ is set as 1. For head and tail model $C_h(\cdot)$, $C_t(\cdot)$, we adopt neural networks with structure $[d, 1024, 800, 512, k]$ for CUB dataset and $[d, 256, 64, k]$ for other datasets. The module of correlation mining aggregator and semi-supervised regularization are selected according to the validation performance. For multiplication aggregator, $C_R(\cdot)$ is a network with $[k^2, k]$ and for addition aggregator, $C_R(\cdot)$ is $[2 \times k, k]$. For all compared methods, we also conduct parameter selection by performing evaluation metrics on the validation set. To reduce statistical variability, all reported results are averaged over 10 independent runs.

4.2 DiDi Liability Judgment Task

We apply our approach to a real-world industrial application, i.e., the liability judgment task from DiDi, one of the largest mobility technology platforms that offer peer-to-peer ride-sharing services in the world. Once passengers enter start location and destination, the platform will match a driver nearby to pick up the passenger. If the passenger complains about the driver to the platform after the order is finished, in order to protect the rights of drivers and passengers, the platform needs to judge whether the driver is really responsible and predicts all related decidendi reasons to make the decision convincing.

The DiDi ride-sharing liability judgment dataset contains 25,108 instances constructed from the real ride-sharing orders in Mainland China within the period from November 5th, 2020 to November 23rd, 2020. Each instance in the data set is described with hundreds of features which can be divided into two parts: tabular features and text features. For tabular features, we select 85 top related features using the XGBoost model and then normalized these features into zero mean and unit variance. For text features, we adapt TextCNN [13] to process the after-ride text into 192-dimension feature vectors and a HAN [29] model to process the conversation text into 200-dimension feature vectors. Overall, we got 477-dimension features for each instance. Meanwhile, 6 main decidendi reasons are adopted as the labels for each instance. We adopt the *class-imbalance ratio* [32], while is defined as the averaged binary class imbalance ratio on each label, to evaluate the skewness of the label distribution. The class-imbalance ratio of DiDi liability dataset is 16.90. It is noteworthy that a dataset is regraded as imbalanced as

Table 6: Experimental results (mean±std) on *Yeast* dataset. ↑ (↓) indicates the larger (smaller) the better. The best performance and its comparable performances are bolded.

5% Labeled Instances									
Methods	Hamming Loss ↓	Ranking Loss ↓	One Error ↓	Coverage ↓	Ave. Precision ↑	Macro AUC ↑	Micro AUC ↑	Macro F1 ↑	Micro F1 ↑
FCN	0.282±0.014	0.203±0.007	0.266±0.014	0.492±0.012	0.721±0.009	0.601±0.020	0.793±0.010	0.370±0.017	0.536±0.022
CAMEL	0.281±0.012	0.220±0.011	0.291±0.023	0.519±0.012	0.709±0.013	0.602±0.017	0.781±0.011	0.381±0.012	0.539±0.018
DBL	0.282±0.013	0.205±0.009	0.291±0.053	0.486±0.012	0.713±0.018	0.600±0.023	0.789±0.015	0.370±0.018	0.535±0.021
DBL+NT	0.281±0.013	0.206±0.007	0.281±0.032	0.491±0.010	0.717±0.012	0.601±0.020	0.789±0.012	0.373±0.016	0.537±0.021
PL	0.281±0.012	0.201±0.006	0.267±0.018	0.487±0.010	0.722±0.007	0.601±0.018	0.794±0.009	0.372±0.012	0.537±0.017
DRML	0.298±0.014	0.215±0.009	0.301±0.022	0.507±0.014	0.708±0.012	0.596±0.012	0.779±0.009	0.387±0.013	0.529±0.015
DRML+DBL	0.298±0.017	0.220±0.010	0.308±0.024	0.512±0.013	0.706±0.012	**0.603±0.012**	0.775±0.010	**0.388±0.013**	0.525±0.019
Proposal	**0.280±0.014**	**0.197±0.006**	**0.258±0.006**	**0.477±0.012**	**0.726±0.008**	0.600±0.018	**0.797±0.007**	0.371±0.017	**0.540±0.021**
10% Labeled Instances									
Methods	Hamming Loss ↓	Ranking Loss ↓	One Error ↓	Coverage ↓	Ave. Precision ↑	Macro AUC ↑	Micro AUC ↑	Macro F1 ↑	Micro F1 ↑
FCN	0.264±0.009	0.192±0.004	0.259±0.010	0.477±0.004	0.734±0.004	0.626±0.007	0.809±0.005	0.396±0.013	0.562±0.015
CAMEL	0.264±0.008	0.207±0.005	0.280±0.015	0.505±0.007	0.722±0.008	0.627±0.015	0.796±0.005	0.405±0.012	**0.564±0.013**
DBL	0.264±0.007	0.192±0.004	0.268±0.018	0.471±0.007	0.734±0.005	0.627±0.010	0.807±0.004	0.394±0.011	0.561±0.013
DBL+NT	0.263±0.008	0.192±0.005	0.266±0.017	0.471±0.006	0.733±0.005	0.625±0.010	0.807±0.004	0.396±0.011	0.562±0.014
PL	0.263±0.008	0.191±0.004	0.260±0.011	0.475±0.005	0.734±0.005	0.626±0.009	0.809±0.004	0.394±0.009	0.561±0.013
DRML	0.301±0.014	0.210±0.010	0.297±0.021	0.499±0.012	0.716±0.013	0.622±0.014	0.791±0.010	**0.410±0.015**	0.537±0.016
DRML+DBL	0.286±0.014	0.214±0.007	0.303±0.015	0.505±0.007	0.713±0.010	0.625±0.011	0.786±0.008	0.402±0.012	0.539±0.019
Proposal	**0.261±0.007**	**0.186±0.003**	**0.249±0.003**	**0.462±0.006**	**0.739±0.003**	**0.629±0.007**	**0.811±0.003**	0.396±0.006	**0.564±0.010**

long as the ratio is grater than 2, therefore, DiDi liability dataset is a highly class imbalanced dataset.

Experimental results are reported in Table 3. It can be seen that the SOTA multi-label learning method CAMEL does not achieve performance improvement over the baseline FCN method. The class imbalanced multi-label learning methods, e.g., DBL, also perform even worse than the baseline method. The main reason is these methods rely on a sufficient number of labeled data and could not work well in the semi-supervised scenario. The semi-supervised multi-label learning methods, e.g., DRML, perform badly on multiple metrics. The main reason is they can not handle the class-imbalance problem. Moreover, simply combines the advantages of semi-supervised and class-imbalanced methods also could not work well. In contrast, our proposal Limi achieves clearly better performance than compared methods on almost every performance metric. These demonstrate the effectiveness of our proposal on real-world industrial tasks.

4.3 Image Annotation Task

We also conducted experiments on various benchmark tasks, the first we conducted is the image annotation task. We use the CUB data set[1], which is the benchmark data set for multi-label image annotation. The CUB dataset contains 10,240 images. The dataset contains 200 birds and the label information can be described by a 312-dimensional vector. The imbalance ratio is 57.72. A pre-trained VGG Networks [20] based on ImageNet is adopted as the feature extractor.

Results are shown in Table 4. From the results, we can see that other compared SOTA methods, e.g., DBL, DRML, suffer performance degradation problems compared with the baseline FCN method. While our proposal achieves good performance that improves clearly over compared methods. These results verify the effectiveness of our proposal.

4.4 Text Categorization Task

Text categorization is another important machine learning task. For Text categorization, we adopt the BibTex data set[2] which is collected from a social bookmarking and publication-sharing system [12]. The recommender should efficiently propose a relevant set of tags to the user when the user submits a new item (BibTeX) into the system. The Bibtex data set includes 7,395 instances and each instance is expressed with 1,836 features. The label information is described by a 159-dimensional vector. The imbalance ratio is 32.25.

Results for BibTex dataset are shown in Table 5. From the results, we can see that the SOTA multi-label learning method CAMEL performs well on multiple performance metrics, especially on macro/micro F1 score and macro/micro AUC. Meanwhile, our proposal still achieves the best performance on more than a half of metrics. These also demonstrate the effectiveness of our proposal Limi framework.

4.5 Gene Function Analysis Task

The last task is to predict the gene function classes of the Yeast Saccharomyces[3], which is one of the best-studied organisms. The Yeast data set [7] is a gene function classification with 2,417 instances

[1] http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

[2] http://mulan.sourceforge.net/datasets-mlc.html

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html

and 14 class labels. Each gene is expressed with 103 microarray expression features. The imbalance ratio of Yeast dataset is 2.78.

Results presented in Table 6 show that our proposal LIMI could achieve highly competitively performance with compared methods. Overall, these empirical results clearly show that the advantage of our proposal which is able to address the imbalanced label distribution, explore the label correlation and exploit the unlabeled data to enhance the performance.

5 CONCLUSION

In this paper, we study the problem of learning from imbalanced and incomplete supervision, which accommodates many real-world applications and has rarely been studied before. We have proposed a systemic framework LIMI that addresses the imbalanced supervision by separating the multiple labels into head labels and tail labels. The prediction results obtained by the head and tail model then input into a novel correlation network to explore the label correlation knowledge. Moreover, the proposed framework can be flexibly incorporated with different semi-supervised learning strategies to further exploit the unlabeled instances. Extensive empirical results on real-world DiDi liability judgment tasks and various benchmark datasets demonstrate that LIMI perform clearly better than many competitive methods. Overall, the proposal is flexible, general, and effective to learn from imbalanced and incomplete supervision.

In some applications, it may be difficult to tune the modules in LIMI. We will consider extending this work with Automated Machine Learning (AutoML) [30] to automatically choose the best sub-modules for different tasks in the future.

6 ACKNOWLEDGMENTS

This research was supported by the National Key R&D Program of China (2017YFB1001903), the National Science Foundation of China (61921006, 61772262), and the Collaborative Innovation Center of Novel Software Technology and Industrialization. We would like to thank Prof. Zhi-Hua Zhou who provided valuable suggestions to this work.

REFERENCES

[1] Hakan Cevikalp, Burak Benligiray, and Ömer Nezih Gerek. 2020. Semi-supervised robust deep neural networks for multi-label image classification. *Pattern Recognition* 100 (2020), 107164.

[2] Francisco Charte, Antonio J. Rivera, María José del Jesus, and Francisco Herrera. 2014. MLeNN: A First Approach to Heuristic Multilabel Undersampling. In *Intelligent Data Engineering and Automated Learning*, Vol. 8669. 1–9.

[3] Francisco Charte, Antonio J. Rivera, María José del Jesus, and Francisco Herrera. 2015. Addressing Imbalance in Multilabel Classification: Measures and Random Resampling Algorithms. *Neuroomputing* 163 (2015), 3–16.

[4] Francisco Charte, Antonio J. Rivera, María José del Jesus, and Francisco Herrera. 2015. MLSMOTE: Approaching Imbalanced Multilabel Learning through Synthetic Instance Generation. *Knowledge Based System* 89 (2015), 385–397.

[5] Zachary Alan Daniels and Dimitris N. Metaxas. 2017. Addressing Imbalance in Multi-Label Classification Using Structured Hellinger Forests. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 1826–1832.

[6] Hao-Chen Dong, Yu-Feng Li, and Zhi-Hua Zhou. 2018. Learning From Semi-Supervised Weak-Label Data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 2926–2933.

[7] André Elisseeff and Jason Weston. 2002. A Kernel Method for Multi-Labelled Classification. In *Advances in Neural Information Processing Systems*. 681–687.

[8] Lei Feng, Bo An, and Shuo He. 2019. Collaboration based Multi-Label Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 3550–3557.

[9] Lan-Zhe Guo, Feng Kuang, Zhang-Xun Liu, Yu-Feng Li, Nan Ma, and Xiao-Hu Qie. 2020. IWE-Net: Instance Weight Network for Locating Negative Comments and its application to improve Traffic User Experience. In *Proceedings of the 34nd AAAI Conference on Artificial Intelligence*. 4052–4059.

[10] Lan-Zhe Guo, Tao Han, and Yu-Feng Li. 2019. Robust Semi-supervised Representation Learning for Graph-Structured Data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 131–143.

[11] Lan-Zhe Guo, Zhen-Yu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. 2020. Safe Deep Semi-Supervised Learning for Unseen-Class Unlabeled Data. In *Proceedings of the 37th International Conference on Machine Learning*. 3897–3906.

[12] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2008. Multi-Label Text Classification for Automated Tag Suggestion. In *Proceedings of the ECML/PKDD*, Vol. 18. 5.

[13] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 1746–1751.

[14] Dong-Hyun Lee. 2013. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. In *Workshop on Challenges in Representation Learning, ICML*.

[15] Joffrey L. Leevy, Taghi M. Khoshgoftaar, Richard A. Bauder, and Naeem Seliya. 2018. A Survey on Addressing High-Class Imbalance in Big Data. *Journal of Big Data* 5 (2018), 42.

[16] Yu-Feng Li, Lan-Zhe Guo, and Zhi-Hua Zhou. 2019. Towards Safe Weakly Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 1 (2019), 334–346.

[17] Yi Liu, Rong Jin, and Liu Yang. 2006. Semi-supervised Multi-label Learning by Constrained Non-negative Matrix Factorization. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence*. 421–426.

[18] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. 2019. Large-Scale Long-Tailed Recognition in an Open World. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2537–2546.

[19] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. 2018. Realistic Evaluation of Deep Semi-Supervised Learning algorithms. In *Advances in Neural Information Processing Systems*. 3235–3246.

[20] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of the 3rd International Conference on Learning Representations*.

[21] Muhammad Atif Tahir, Josef Kittler, and Ahmed Bouridane. 2012. Multilabel Classification using Heterogeneous Ensemble of Multi-Label Classifiers. *Pattern Recognition Letters* 33, 5 (2012), 513–523.

[22] Qiaoyu Tan, Yanming Yu, Guoxian Yu, and Jun Wang. 2017. Semi-Supervised Multi-Label Classification using Incomplete Label Information. *Neurocomputing* 260 (2017), 192–202.

[23] Lichen Wang, Yunyu Liu, Can Qin, Gan Sun, and Yun Fu. 2020. Dual Relation Semi-Supervised Multi-Label Learning. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. 6227–6234.

[24] Tong Wei, Lan-Zhe Guo, Yu-Feng Li, and Wei Gao. 2018. Learning Safe Multi-Label Prediction for Weakly Labeled Data. *Machine Learning* 107, 4 (2018), 703–725.

[25] Le Wu and Min-Ling Zhang. 2013. Multi-Label Classification with Unlabeled Data: An Inductive Approach. In *Proceedings of the Asian Conference on Machine Learning, 2013*, Vol. 29. 197–212.

[26] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. 2020. Distribution-Balanced Loss for Multi-Label Classification in Long-Tailed Datasets. In *European Conference on Computer Vision*. 162–178.

[27] Xi-Zhu Wu and Zhi-Hua Zhou. 2017. A Unified View of Multi-Label Performance Measures. In *Proceedings of the 34th International Conference on Machine Learning*. 3780–3788.

[28] Yuying Xing, Guoxian Yu, Carlotta Domeniconi, Jun Wang, and Zili Zhang. 2018. Multi-Label Co-Training. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2882–2888.

[29] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1480–1489.

[30] Quanming Yao, Mengshuo Wang, Hugo Jair Escalante, Isabelle Guyon, Yi-Qi Hu, Yu-Feng Li, Wei-Wei Tu, Qiang Yang, and Yang Yu. 2018. Taking Human out of Learning Applications: A Survey on Automated Machine Learning. *CoRR* abs/1810.13306 (2018).

[31] Wang Zhan and Min-Ling Zhang. 2017. Inductive Semi-Supervised Multi-Label Learning with Co-Training. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1305–1314.

[32] Min-Ling Zhang, Yu-Kun Li, and Xu-Ying Liu. 2015. Towards Class-Imbalance Aware Multi-Label Learning. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. 4041–4047.

[33] Min-Ling Zhang and Zhi-Hua Zhou. 2013. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26, 8 (2013), 1819–1837.