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

Lan-Zhe Guo†, Zhi Zhou†, Jie-Jing Shao†, Qi Zhang2, Feng Kuang2, Gao-Le Li2, Zhang-Xun Liu2, Guo-Bin Wu2, Nan Ma2, Qun Li2, Yu-Feng Li§

1National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
{guolz,zhouz,shaojj,liyf}@lamda.nju.edu.cn
2Didi Chuxing, Beijing, China
{zhangqi,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 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 judgement

†Corresponding author
§Contribute to this work equally

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD ’21, August 14–18, 2021, Virtual Event, Singapore
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8332-5/21/08... $15.00
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 [10], 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
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 the 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 issue 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 ride-sharing liability judgment task 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.

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.
in each label. MLenn [2] is an undersampling algorithm based on the ENN (Edited Nearest Neighbor) rule. To reduce the risk of over-fitting 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 datasets. 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 utilize labeled as well as unlabeled data simultaneously [34], [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 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 on unlabeled instances. [23] introduced a DRML approach which 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 ignores the inherent class imbalance problem in multi-label data and could suffer severe performance degradation problem.

### 3 LEARNING FROM IMBALANCED AND IMCOMPLETE 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 $X = \mathbb{R}^d$ be a $d$-dimensional input feature space and $Y = \{0, 1\}^k$ a $k$-dimensional label space. Given $D_l = \{(x_l, y_l)\}_{1 \leq i \leq n}$ be the labeled training data containing $n$ instances. $X_l \in \mathbb{R}^d$ and $Y_l \in \{0, 1\}^k$ are the labeled instances, $\hat{Y}(x_l)$ represent the aggregated result of $C_h(x_l)$ and $C_t(x_l)$, where $C_h(\cdot)$ and $C_t(\cdot)$ are learned functions. $f(x_l)$ is the final prediction of the Limi framework.

#### Table 1: Summary of Notations.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>Number of labeled instances</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of unlabeled instances</td>
</tr>
<tr>
<td>$k$</td>
<td>Number of labels</td>
</tr>
<tr>
<td>$x \in X = \mathbb{R}^d$</td>
<td>Feature vector of instances</td>
</tr>
<tr>
<td>$y \in Y = {0, 1}^k$</td>
<td>Ground-truth label of instances</td>
</tr>
<tr>
<td>$\hat{y} \in Y$</td>
<td>Predicted label of instances</td>
</tr>
<tr>
<td>$R_l$</td>
<td>Aggregated result of $C_h(x_l)$ and $C_t(x_l)$</td>
</tr>
<tr>
<td>$f : X \rightarrow Y$</td>
<td>Learned function</td>
</tr>
<tr>
<td>$C_h(x_l) \in [0, 1]^k$</td>
<td>Predicted probabilities of head model</td>
</tr>
<tr>
<td>$C_t(x_l) \in [0, 1]^k$</td>
<td>Predicted probabilities of tail model</td>
</tr>
<tr>
<td>$f(x_l) \in [0, 1]^k$</td>
<td>Predicted probabilities of Limi framework</td>
</tr>
</tbody>
</table>

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

\[
\min_{C_h} \ell_h(C_h(X), Y) = BCE(C_h(X), Y) \tag{1}
\]

\( \ell_h \) indicates the loss function for head model and

\[
BCE(C_h(x), y) = \frac{1}{k} \sum_{j=1}^{k} [y^j \log(C_h(x)_{ij}) + (1-y^j) \log(1-C_h(x)_{ij})] \tag{2}
\]

where \( C_h(x) \in \{0, 1\}^k \) is the predicted label vector for instance \( x \) of head model, and \( C_h(x)_{ij} \) indicates the predicted probability that whether \( x_i \) is relevant with label \( j \), \( y^j \) is the true label of instance \( x_i \) 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 \( y^j_i = 1 \), the expectation of class-level sampling frequency can be calculated as:

\[
P^j_i(x_i) = \frac{1}{k} \frac{1}{n_j} \tag{3}
\]

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 its 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:

\[
P^j_i(x_i) = \frac{1}{k} \sum_{j=1}^{k} \frac{1}{n_j} \tag{4}
\]

Correspondingly, we can calculate a re-balancing weight \( r^j_i \) to close the gap between expected sampling times and actual sampling times:

\[
r^j_i = \frac{P^j_i(x_i)}{P^j_i(x)} \tag{5}
\]

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

\[
\ell_t(C_t(X), y) = \frac{1}{nk} \sum_{i=1}^{n} \sum_{j=1}^{k} BCE(x_i, y^j_i) \cdot r^j_i \tag{6}
\]

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:

\[
\min_{C_h, C_t} \ell_h(C_h(X), Y) + \ell_t(C_t(X), Y) \tag{7}
\]

### 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 \( C_h(x) \) and \( C_t(x) \) and obtain

\[
R_i = w_h C_h(x_i) + w_t C_t(x_i) \tag{8}
\]

The obtained \( R_i \in \mathbb{R}^{1 \times k} \) is then transformed into a new label space to obtain the final prediction result \( C_R(w_h C_h(x) + w_t C_t(x)) \).

**Multiplication Aggregator.** We also examine a more complex aggregator by multiplying the transposition of \( C_h(x) \) and \( C_t(x) \) and obtain

\[
R_i = C_h(x_i)^T \times C_t(x_i) \tag{9}
\]

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:

\[
\min_{C_R,C_h,C_t} \ell_R(C_R(X_i, Y_j)) = BCE(C_R(R_i), Y_j) \tag{10}
\]

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 \( S \) that can be defined with RBF kernel, i.e., \( S_{ij} = \exp\left(\frac{-\|x_i - x_j\|^2}{\sigma^2}\right) \). Denote the final predicted probability of our framework for instance \( x \) is \( f(x) \), i.e., \( f(x) = C_R(C_h(x), C_t(x)) \), the unsupervised regularizer can be written as:

\[
\ell_u(X) = \frac{1}{2} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} (S_{ij})(f(x_i) - f(x_j))^2 \tag{12}
\]

\[
f(X)^T L f(X)
\]

where \( X = [X_1, X_u] \), \( L = D - S \) is the Laplacian matrix of \( S \) and \( D \) 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, 9].

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:

\[
\ell_u(X) = \sum_{i=1}^{n+m} \sum_{j=1}^{k} H(2 * |f(x_i)_j - 0.5|) \tag{13}
\]

**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 LIMIT contains three modules \( C_h(\cdot) \), \( C_t(\cdot) \) and \( C_R(\cdot) \), which are jointly optimized by minimizing the following objective:

\[
\min_{C_h,C_t,C_R} \ell_h + \ell_t + \ell_R + \lambda L_u \tag{14}
\]

where \( \lambda \) 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 LIMIT 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>
<thead>
<tr>
<th>Measure</th>
<th>Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hamming loss</td>
<td>( \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} C_i \hat{y}<em>{ij} - y</em>{ij} )</td>
</tr>
<tr>
<td>Ranking Loss</td>
<td>( \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{</td>
</tr>
<tr>
<td>One Error</td>
<td>( \frac{1}{N} \sum_{i=1}^{N} \left( \text{argmax} f(x_i) \neq Y_i^+ \right) )</td>
</tr>
<tr>
<td>Coverage</td>
<td>( \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{</td>
</tr>
<tr>
<td>Macro AUC</td>
<td>( \frac{1}{K} \sum_{k=1}^{K} \frac{</td>
</tr>
<tr>
<td>Micro AUC</td>
<td>( \frac{</td>
</tr>
<tr>
<td>Macro F1</td>
<td>( \frac{1}{K} \sum_{k=1}^{K} \frac{2 \sum_{i=1}^{N}</td>
</tr>
<tr>
<td>Micro F1</td>
<td>( \frac{2 \sum_{j=1}^{N}</td>
</tr>
</tbody>
</table>

**Table 2: Definition of 9 Multi-Label Performance Metrics**
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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Hamming Loss</th>
<th>Ranking Loss</th>
<th>One Error</th>
<th>Coverage</th>
<th>Ave. Precision</th>
<th>Macro AUC</th>
<th>Micro AUC</th>
<th>Macro F1</th>
<th>Micro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN</td>
<td>0.160±0.008</td>
<td>0.100±0.001</td>
<td>0.237±0.001</td>
<td>0.121±0.001</td>
<td>0.839±0.001</td>
<td>0.588±0.012</td>
<td>0.885±0.004</td>
<td>0.239±0.014</td>
<td>0.584±0.016</td>
</tr>
<tr>
<td>CAMEL</td>
<td>0.159±0.006</td>
<td>0.115±0.003</td>
<td>0.243±0.002</td>
<td>0.135±0.002</td>
<td>0.831±0.001</td>
<td>0.593±0.006</td>
<td>0.878±0.002</td>
<td>0.239±0.006</td>
<td>0.585±0.017</td>
</tr>
<tr>
<td>DBL</td>
<td>0.160±0.008</td>
<td>0.100±0.001</td>
<td>0.236±0.001</td>
<td>0.121±0.001</td>
<td>0.840±0.001</td>
<td>0.586±0.017</td>
<td>0.885±0.005</td>
<td>0.237±0.017</td>
<td>0.583±0.015</td>
</tr>
<tr>
<td>DBL+NT</td>
<td>0.157±0.006</td>
<td>0.100±0.001</td>
<td>0.237±0.002</td>
<td>0.121±0.001</td>
<td>0.839±0.001</td>
<td>0.592±0.009</td>
<td>0.884±0.003</td>
<td>0.243±0.007</td>
<td>0.591±0.013</td>
</tr>
<tr>
<td>DRML</td>
<td>0.161±0.003</td>
<td>0.108±0.002</td>
<td>0.264±0.003</td>
<td>0.127±0.002</td>
<td>0.825±0.002</td>
<td>0.524±0.016</td>
<td>0.870±0.005</td>
<td>0.217±0.007</td>
<td>0.579±0.026</td>
</tr>
<tr>
<td>DRML+DBL</td>
<td>0.164±0.009</td>
<td>0.116±0.008</td>
<td>0.276±0.019</td>
<td>0.135±0.006</td>
<td>0.816±0.011</td>
<td>0.533±0.011</td>
<td>0.858±0.010</td>
<td>0.221±0.005</td>
<td>0.574±0.022</td>
</tr>
<tr>
<td>Proposal</td>
<td>0.152±0.010</td>
<td>0.099±0.001</td>
<td>0.236±0.002</td>
<td>0.120±0.001</td>
<td>0.840±0.001</td>
<td>0.589±0.025</td>
<td>0.889±0.004</td>
<td>0.232±0.014</td>
<td>0.597±0.019</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Methods</th>
<th>Hamming Loss</th>
<th>Ranking Loss</th>
<th>One Error</th>
<th>Coverage</th>
<th>Ave. Precision</th>
<th>Macro AUC</th>
<th>Micro AUC</th>
<th>Macro F1</th>
<th>Micro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN</td>
<td>0.155±0.006</td>
<td>0.098±0.001</td>
<td>0.237±0.002</td>
<td>0.120±0.001</td>
<td>0.841±0.001</td>
<td>0.607±0.016</td>
<td>0.889±0.004</td>
<td>0.249±0.015</td>
<td>0.596±0.013</td>
</tr>
<tr>
<td>CAMEL</td>
<td>0.156±0.003</td>
<td>0.112±0.001</td>
<td>0.240±0.002</td>
<td>0.132±0.001</td>
<td>0.834±0.001</td>
<td>0.606±0.005</td>
<td>0.882±0.001</td>
<td>0.248±0.007</td>
<td>0.593±0.009</td>
</tr>
<tr>
<td>DBL</td>
<td>0.156±0.006</td>
<td>0.098±0.002</td>
<td>0.236±0.001</td>
<td>0.120±0.001</td>
<td>0.841±0.001</td>
<td>0.604±0.019</td>
<td>0.889±0.004</td>
<td>0.249±0.013</td>
<td>0.596±0.014</td>
</tr>
<tr>
<td>DBL+NT</td>
<td>0.154±0.006</td>
<td>0.098±0.001</td>
<td>0.236±0.001</td>
<td>0.120±0.001</td>
<td>0.841±0.001</td>
<td>0.601±0.018</td>
<td>0.889±0.004</td>
<td>0.245±0.013</td>
<td>0.599±0.012</td>
</tr>
<tr>
<td>PL</td>
<td>0.154±0.005</td>
<td>0.098±0.001</td>
<td>0.237±0.002</td>
<td>0.120±0.001</td>
<td>0.840±0.001</td>
<td>0.613±0.007</td>
<td>0.891±0.002</td>
<td>0.255±0.007</td>
<td>0.600±0.009</td>
</tr>
<tr>
<td>DRML</td>
<td>0.160±0.006</td>
<td>0.103±0.001</td>
<td>0.255±0.011</td>
<td>0.123±0.001</td>
<td>0.832±0.005</td>
<td>0.530±0.011</td>
<td>0.876±0.004</td>
<td>0.223±0.005</td>
<td>0.584±0.012</td>
</tr>
<tr>
<td>DRML+DBL</td>
<td>0.160±0.008</td>
<td>0.105±0.002</td>
<td>0.252±0.011</td>
<td>0.125±0.002</td>
<td>0.831±0.005</td>
<td>0.533±0.013</td>
<td>0.871±0.004</td>
<td>0.222±0.005</td>
<td>0.585±0.017</td>
</tr>
<tr>
<td>Proposal</td>
<td>0.150±0.005</td>
<td>0.096±0.002</td>
<td>0.234±0.001</td>
<td>0.115±0.002</td>
<td>0.843±0.002</td>
<td>0.614±0.023</td>
<td>0.892±0.010</td>
<td>0.260±0.010</td>
<td>0.599±0.012</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>5% Labeled Instances</th>
<th>10% Labeled Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hamming Loss</td>
<td>Ranking Loss</td>
</tr>
<tr>
<td>FCN</td>
<td>0.025±0.001</td>
<td>0.163±0.006</td>
</tr>
<tr>
<td>CAMEL</td>
<td>0.024±0.001</td>
<td>0.176±0.005</td>
</tr>
<tr>
<td>DBL</td>
<td>0.025±0.001</td>
<td>0.163±0.006</td>
</tr>
<tr>
<td>DBL+NT</td>
<td>0.024±0.001</td>
<td>0.164±0.005</td>
</tr>
<tr>
<td>PL</td>
<td>0.027±0.001</td>
<td>0.182±0.006</td>
</tr>
<tr>
<td>DRML</td>
<td>0.024±0.002</td>
<td>0.343±0.009</td>
</tr>
<tr>
<td>DRML+DBL</td>
<td>0.028±0.002</td>
<td>0.289±0.011</td>
</tr>
<tr>
<td>Proposal</td>
<td>0.024±0.001</td>
<td>0.160±0.007</td>
</tr>
</tbody>
</table>

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_\ell(·), C_t(·), 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_p(·) is a network with [k^2, k] and for addition aggregator, C_a(·) is [2 × 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 online ride-sharing 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 deciden 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 deciden 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
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-imbalance 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*, 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 LMI 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 LMI that address 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 task and various benchmark datasets demonstrate that LMI perform clearly better than many competitive methods. Overall, the proposal is flexible, general, and effective to learn from imbalanced and incomplete supervision.

In LMI, we provide different label aggregators and semi-supervised regularizations. In some applications, it may be difficult to tune these modules. 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 NSFC (61772262), and the Fundamental Research Funds for the Central Universities (14380061). We would like to thank Prof. Zhi-Hua Zhou who provided valuable suggestions to this work.

REFERENCES