Deep Partial Multi-Label Learning with Graph Disambiguation

Haobo Wang⁴, Shisong Yang⁵, Gengyu Lyu⁶*, Weiwei Liu⁶, Tianlei Hu⁴, Ke Chen⁴, Songhe Feng⁶ and Gang Chen⁴

1Zhejiang University
2Beijing University of Technology
3Wuhan University
4Beijing Jiaotong University
5{wanghaobo, htl, chenk, cg}@zju.edu.cn, yangshisong@emails.bjtu.edu.cn, lyugengyu@bjut.edu.cn, liuweiwei863@gmail.com, shfeng@bjtu.edu.cn

Abstract
In partial multi-label learning (PML), each data example is equipped with a candidate label set, which consists of multiple ground-truth labels and other false-positive labels. Recently, graph-based methods, which demonstrate a good ability to estimate accurate confidence scores from candidate labels, have been prevalent to deal with PML problems. However, we observe that existing graph-based PML methods typically adopt linear multi-label classifiers and thus fail to achieve superior performance. In this work, we attempt to remove several obstacles for extending them to deep models and propose a novel deep Partial multi-Label model with grAph-disambiguatioN (PLAIN). Specifically, we introduce the instance-level and label-level similarities to recover label confidences as well as exploit label dependencies. At each training epoch, labels are propagated on the instance and label graphs to produce relatively accurate pseudo-labels; then, we train the deep model to fit the numerical labels. Moreover, we provide a careful analysis of the risk functions to guarantee the robustness of the proposed model. Extensive experiments on various synthetic datasets and three real-world PML datasets demonstrate that PLAIN achieves significantly superior results to state-of-the-art methods.

1 Introduction
The rapid growth of deep learning allows us to tackle increasingly sophisticated problems. Among them, multi-label learning (MLL), which assigns multiple labels to an object, is a fundamental and intriguing task in various real-world applications such as image retrieval [Lai et al., 2016], autonomous vehicles [Chen et al., 2019a], and sentiment analysis [Yilmaz et al., 2021]. Typically, these tasks require precisely labeled data, which is expensive and time-consuming due to the complicated structure and the high volume of the output space.

To address this issue, plenty of works have studied different weakly-supervised settings of MLL, including semi-supervised MLL [Shi et al., 2020], multi-label with missing labels [Durand et al., 2019; Huynh and Elhamifar, 2020], and single-labeled MLL [Xu et al., 2022]. While these learning paradigms mainly battle the rapid growth of data volume, they overlook the intrinsic difficulty and ambiguity in multi-label annotation. For example, it is hard for a human annotator to identify an Alaskan Malamute from a Siberian Husky. One potential solution is to randomly decide the binary label value. However, it may involve uncontrollable label noise that has a negative impact on the training procedure. Another strategy is to leave it missing, but then, we may lose critical information. Recently, partial multi-label learning (PML) [Xie and Huang, 2018] has been proposed to alleviate this problem. In PML, the annotators are allowed to provide a candidate label set that contains all potentially relevant labels as well as false-positive labels.

A straightforward approach to learning from PML data is to regard all candidate labels as valid ones, and then, off-the-shelf multi-label methods can be applied. Nevertheless, the presence of false-positive labels can highly impact predictive performance. To cope with this issue, many PML methods [Xie and Huang, 2018; Yu et al., 2018; He et al., 2019; Sun et al., 2019; Li et al., 2020; Lyu et al., 2020; Yan et al., 2021; Xie and Huang, 2022] have been developed. Amongst them, graph-based PML methods [Wang et al., 2019; Li and Wang, 2020; Xu et al., 2020; Lyu et al., 2020; Zhang and Fang, 2021] have been popular. In practice, graph-based algorithms demonstrate a good ability to obtain relatively precise confidence of candidate labels by aggregating information from the nearest neighbors. However, we observe that most of them adopt linear classifier, which is powerless to deal with complicated tasks and high data volumes.

Therefore, there is an urgent need for developing an effective deep model to boost the performance of graph PML methods. However, there are two main challenging issues. First, existing graph-based PML methods typically require propagating labels over the whole training dataset. It restricts their scalability when employing deep classifiers since the deep models are trained in batch mode. Second, it is well acknowledged that there exist label correlations in multi-label...
datasets, which also helps disambiguate the candidate labels. For example, if labels *Amusement Park* and *Mickey Mouse* are identified as true, it is very likely that label *Disney* is true as well. Thus, it is required to exploit the label correlations to improve the disambiguation ability in a unified framework.

In this work, we propose a novel deep Partial multi-Label model with grAph disambiguatIoN (PLAIN). In PLAIN, we involve both instance- and label-level similarities to disambiguate the candidate labels set, which simultaneously leverages instance smoothness as well as label correlations. Then, we build a deep multi-label classifier to fit the relatively credible pseudo-labels generated through disambiguation. Instead of separately training in two stages like previous works [Wang et al., 2019; Zhang and Fang, 2021], we propose an efficient optimization framework that iteratively propagates the labels and trains the deep model. Moreover, we give a careful analysis of the risk functions such that proper risk functions are chosen for improved robustness. Empirical results on nine synthetic datasets and three real-world PML datasets clearly verify the efficiency and effectiveness of our method. More theoretical and empirical results can be found in Appendix1.

2 Related Works

**Multi-Label Learning.** MLL [Liu et al., 2022; Wei et al., 2022] aims at assigning each data example multiple binary labels simultaneously. An intuitive solution, the one-versus-all method (OVA) [Zhang and Zhou, 2014], is to decompose the original problem into a series of single-labeled classification problems. However, it overlooks the rich semantic information and dependencies among labels, and thus fails to obtain satisfactory performance. To solve this problem, many MLL approaches are proposed, such as embedding-based methods [Yu et al., 2014; Shen et al., 2018], classifier chains [Read et al., 2011], tree-based algorithms [Wei et al., 2021], and deep MLL models [Yeh et al., 2017; Chen et al., 2019b; Bai et al., 2020]. Nevertheless, these methods typically require a large amount of fully-supervised training data, which is expensive and time-consuming in MLL tasks. To this end, some works have studied different weakly-supervised MLL settings. For example, semi-supervised MLL [Niu et al., 2019; Shi et al., 2020] leverages information from both fully-labeled and unlabeled data. Multi-label with missing labels [Durand et al., 2019; Wei et al., 2018; Xu et al., 2022] allows providing a subset of ground-truth. In noisy MLL [Veit et al., 2017], the binary labels may be flipped to the wrong one. Nevertheless, though these settings relieve the burden of multi-label annotating, they ignore the inherent difficulty of labeling, i.e. the labels can be ambiguous.

**Partial Multi-Label Learning.** PML learns from a super-set of ground-truth labels, where multiple labels are possible to be relevant. To tackle the ambiguity in the labels, the pioneering work [Xie and Huang, 2018] estimates the confidence of each candidate label being correct via minimizing confidence-weighted ranking loss. Some works [Yu et al., 2018; Sun et al., 2019; Li et al., 2020; Gong et al., 2022] recover the ground-truth by assuming the true label matrix is low-rank. Recently, graph-based methods [Wang et al., 2019; Xu et al., 2020; Lyu et al., 2020; Xie and Huang, 2020; Zhang and Fang, 2021; Xu et al., 2021; Tan et al., 2022] have attracted much attention from the community due to the good disambiguation ability. For instance, PARTICLE [Zhang and Fang, 2021] identifies the credible labels through an iterative label propagation procedure and then, applies pair-wise ranking techniques to induce a multi-label predictor. Although graph-based models have shown promising results, we observe that most of them adopt linear classifiers, which significantly limits the model expressiveness. Besides, they typically separate the graph disambiguation and model training into two stages, which makes the multi-label classifier error-prone; in contrast, the proposed PLAIN is an end-to-end framework.

Some studies have also developed deep PML models, such as adversarial PML model [Yan and Guo, 2019] and mutual teaching networks [Yan et al., 2021]. However, it remains urgent to combine the graph and deep models together such that the model enjoys high expressiveness as well as good disambiguation ability. It should also be noted that the recent popular graph neural networks (GNN) model [Wu et al., 2021] is not suitable for PML. The reason is that GNN aims at better representation learning (at an instance-level), while in PML, the graph structure is used to alleviate the label ambiguity.

3 Method

We denote the $d$-dimensional feature space as $\mathcal{X} \subset \mathbb{R}^d$, and the label space as $\mathcal{Y} = \{1, 2, \ldots, L\}$ with $L$ class labels. The training dataset $\mathcal{D} = \{(x_i,S_i)\}_{1 \leq i \leq n}$ contains $n$ examples, where $x_i \in \mathcal{X}$ is the instance vector and $S_i \subset \mathcal{Y}$ is the candidate label set. Without loss of generality, we assume that the instance vectors are normalized such that $\|x_i\|_2 = 1$. Besides, we denote $\tilde{S}_i \subset \mathcal{Y}$ as the ground-truth label set, which is a subset of $S_i$. For simplicity, we define $y_i, \tilde{y}_i \in \{0,1\}^L$ as the binary vector representations of $S_i$ and $\tilde{S}_i$. When considering the whole dataset, we also denote the candidate label matrix by $Y = [y_1, \ldots, y_n]^\top \in \mathbb{R}^{n \times L}$.

In what follows, we first introduce the learning target of our model that integrates label correlations to boost disambiguation ability. Then, we provide an efficient optimization framework for our deep propagation network.

3.1 Learning Objective

Our ultimate goal is to estimate a multi-label predictor $f: \mathcal{X} \mapsto [0,1]^L$ from $\mathcal{D}$. In this work, we assume that its output $\hat{y} = f(x)$ satisfies the following properties,

1. $\hat{y}$ is close to the candidate vector $y$;
2. $\hat{y}$ satisfies the instance-level similarity. That is, if two instances are close to each other, then their label vectors are likely to be close too;
3. $\hat{y}$ satisfies the label-level similarity. That is, if two labels co-occur in the training data frequently, then they tend to co-exist at many instances.

These properties coincide with our goals. The first property is natural since the only thing we can trust is the candidate.

1https://arxiv.org/abs/2305.05882
label set. The second one has been widely adopted to disambiguate the ground-truth label from the false-positive labels. The last one exploits the label correlations to further promote the performance. In this paper, we adopt a graph-based model to achieve these goals. Generally speaking, the graph structure can be collected in various ways. One direct strategy is to use prior knowledge. For instance, for data mining tasks, social networks [Qiu et al., 2018] can be used to depict instance similarity. The label similarity can also be estimated using an external knowledge graph [Lee et al., 2018].

However, we notice that there is no available graph structure in existing public PML datasets. Hence, we propose to construct the instance graph $A^x \in \mathbb{R}^{n \times n}$ and the label graph $A^y \in \mathbb{R}^{L \times L}$ in a data-driven way. To obtain $A^x$, we adopt a monomial kernel [Iscen et al., 2017] as the distance metric and calculate a sparse instance affinity matrix $S \in \mathbb{R}^{n \times n}$ with elements,

$$s_{ij} := \begin{cases} 
\max(x_i^T x_j, 0)^\rho, & \text{if } i \neq j \text{ and } j \in \mathcal{N}_k(x_i) \\
0, & \text{otherwise}
\end{cases}$$

Here $\mathcal{N}_k(x_i)$ saves the indices of $k$-nearest neighbors of $x_i$. $\rho > 0$ controls the weights of similarity. Then, we get a symmetrized instance graph by $A^x = S + S^\top$.

Inspired by HALE [Lyu et al., 2020], we build the label graph $A^y$ by means of the label co-occurrence statistics from the training data, whose elements are given by,

$$a_{ij} := \frac{\sum_{k=1}^n 1(\hat{y}_{ki} = 1 \land y_{kj} = 1)}{\sum_{k=1}^n 1(\hat{y}_{ki} = 1) + \sum_{k=1}^n 1(y_{kj} = 1)},$$

where $1(\cdot)$ is an indicator function returning 1 if the event is true and 0 otherwise. It is worth noting that HALE simply calculates the average co-occurrence by dividing the instance number, which overlooks the existence of false-positive labels. In our method, we penalize it by the counts of instances that are assigned the labels as the candidates. In other words, if one label is assigned to too many instances, its co-occurrence statistics are less reliable.

Next, we impose the instance-level similarity property on the model output by means of a Laplacian regularizer,

$$\mathcal{J}_x(\hat{Y}) = \sum_{i,j=1}^n a_{ij}^x ||\frac{\hat{y}_i}{\sqrt{d_i}} - \frac{\hat{y}_j}{\sqrt{d_j}}||^2 = \text{tr}(\hat{Y}^\top L_x \hat{Y}), \quad (1)$$

where $\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_n)^\top$ is the predicted label matrix. $L_x = I - D_x^{-\frac{1}{2}} A^x D_x^{-\frac{1}{2}}$ is the Laplacian of instance graph. $D_x = \text{diag}(d_1, \ldots, d_n)$ is a diagonal matrix with $d_i = \sum_{j=1}^n a_{ij}$. The objective in Eq. (1) enforces two data examples to have similar predicted labels if they have a large affinity score. Similarly, we can introduce the label-level similarity by,

$$\mathcal{J}_y(\hat{Y}) = \text{tr}(\hat{Y} L_y \hat{Y}^\top), \quad L_y = I - D_y^{-\frac{1}{2}} A^y D_y^{-\frac{1}{2}}.$$

Finally, we involve a candidate consistency term to get the following objective,

$$\mathcal{J}(\hat{Y}, Y) = \frac{\eta}{2} ||\hat{Y} - Y||_F^2 + \frac{\alpha}{2} \mathcal{J}_x(\hat{Y}) + \frac{\beta}{2} \mathcal{J}_y(\hat{Y}),$$

where $|| \cdot ||_F$ is the Frobenius norm. $\eta, \alpha,$ and $\beta$ are all positive hyperparameters.

In summary, the main goal of our objective is to utilize the instance- and label-level similarity to recover reliable label confidences and address the problem of false positives, while simultaneously enjoying an extra performance improvement by exploiting label correlations.

### 3.2 Deep Propagation Network

If a linear model is adopted, the parameters can be directly obtained by optimizing $\mathcal{J}(f(X), Y)$ with gradient descent. However, in deep models, it is not the case. When we update the model using a mini-batch of data, the instance-level manifold regularizer can involve unexpected gradient updating on the nearest neighbors of these data. Then, the training time will increase rapidly. To remedy this problem, we propose an efficient optimization framework for training a deep model from PML data. Specifically, we maintain a smooth pseudo-label matrix $\tilde{Z}$, which minimizes the propagation error while being close to the model output $\hat{Y}$. In the sequel, we elaborate the training procedure of our model. The training procedure is also summarized in Algorithm 1.
**Propagation Step**
At $t$-th epoch, we first propagate the label information from candidates $Y$ and the model output $\hat{Y}$ to the intermediate ones. In graph-based methods, the output label is required to be smooth enough. Therefore, we use the Frobenius norm as the measure to ensure the closeness of $Z$ and $\hat{Y}$. Formally, we estimate $Z$ by,

$$
\min_Z L_{\text{LP}}(Z) = \frac{1}{2} \| Z - \hat{Y} ||_F^2 + J(Z, Y).
$$

(2)

We can derive its gradient by,

$$
\frac{\partial L_{\text{LP}}}{\partial Z} = (1 + \eta) Z + \alpha L_x Z + \beta Z L_y - (\hat{Y} + \eta Y).
$$

(3)

which is followed by the matrix properties that $\frac{\partial ||Z - Y||_F^2}{\partial Z} = 2(Z - Y)$ and $\frac{\partial (Z^T L_x Z)}{\partial Z} = (L_x + L_x^T) Z$. We have $L_x + L_y^T = 2 L_x$ because $L_x$ is symmetric. The optimal condition $\frac{\partial L_{\text{LP}}}{\partial Z} = 0$ typically requires solving a Sylvester system, which can be time-consuming. Therefore, we solve this problem by gradient descent $Z = Z - \gamma \frac{\partial L_{\text{LP}}}{\partial Z}$, where $\gamma > 0$ is the learning rate. Note that in MLL datasets, negative labels usually dominate, which makes all the pseudo-labels tend to concentrate at a small value. To cope with this issue, we further perform column-wise min-max normalization on $Z_t$ to obtain a balanced output.

**Model Updating Step**
Given an instance vector $x$, we first feed it into a deep neural model $\hat{y} = f(x; \theta)$. Then, we update the model parameters to fit the pseudo-labels,

$$
\min_{\theta} L_{\text{Deep}}(\hat{y}, Z) = \frac{1}{n} \sum_{i=1}^{n} l(\hat{y}_i, z_i),
$$

(4)

where $l(\cdot, \cdot)$ is the loss function. Recall that in Eq. (2), we enforce $Z$ to be close to $\hat{Y}$ using the Frobenius norm. Hence, a natural choice of loss is to reuse the mean squared error (MSE), which is adopted by most graph-based PML algorithms [Wang et al., 2019; Xu et al., 2020; Zhang and Fang, 2021]. However, in PML tasks, the disambiguated labels can still be noisy and the MSE loss leads to heavy overfitting. This is largely overlooked by previous works.

To tackle this problem, we shall have a closer look at the risk function. While MSE penalize too much on prediction mistakes, other robust surrogate losses can be considered, such as mean absolute error (MAE). However, we notice that MAE violates our smooth prediction assumption. To this end, we adopt the widely-used binary cross-entropy loss (BCE) for multi-label learning,

$$
l(\hat{y}_i, z_i) = \sum_{j=1}^{L} z_{ij} \log(\sigma(\hat{y}_{ij})) - (1 - z_{ij}) \log(1 - \sigma(\hat{y}_{ij})),
$$

where $\sigma(\cdot)$ is the sigmoid function. On one hand, BCE loss approximately grows linearly for largely negative values, which makes it less sensitive to outliers. On the other hand, BCE outputs relatively smooth output such that the smoothness is preserved.

In effect, we can go further on the risk. A plethora of works [Ghosh et al., 2017; Feng et al., 2020] has demonstrated that BCE is still less robust since it is unbounded. Let us consider the probabilistic output $\sigma(\hat{y}_i)$ of the deep model. Then the BCE loss is equivalent to a negative log loss on $\sigma(\hat{y}_i)$. If the predicted probability of the label is close to zero, the loss can grow quickly to infinity. Previous works have suggested that bounded loss on the probabilistic output is theoretically more robust than unbounded ones. One common choice of bounded loss is the probabilistic MSE loss (PMSE), i.e. $l(\hat{y}_i, z_i) = ||\sigma(\hat{y}_i) - z_i||_2^2$. In that case, the training criterion is consistent with our propagation error. Empirically, both PMSE and BCE loss perform well, and thus, we instantiate our model using BCE loss when comparing it with other methods. In Section 4.4, we present a comparison between these losses, which clearly verifies our conjectures.

**Fast Implementation.** To efficiently build the instance graph from training data, we employ a quick similarity search python package Faiss [Johnson et al., 2017] to find the nearest neighbors. The sparse matrix multiplication $L_x Z$ can be efficiently implemented by the Scipy package. In Appendix A, we provide a detailed analysis of the complexity and convergence properties to show the efficacy of our method.

**4 Experiments**
In this section, we present the main empirical results of PLAIN compared with state-of-the-art baselines. More experimental results are shown in Appendix B.

### 4.1 Datasets
We employ a total of twelve synthetic as well as real-world datasets for comparative studies. Specifically, the synthetic ones are generated from the widely-used multi-label learning datasets. A total of nine benchmark multi-label datasets² are used for synthetic PML datasets generation, including Emotions, Birds, Medical, Image, Bibtext, Corel5K, Eurlex-dc.

²http://mulan.sourceforge.net/datasets-mlc.html
Eurlex-sm, NUS-WIDE. Note that NUS-WIDE is a large-scale dataset, and we preprocess it as in [Lyu et al., 2020]. For each data example in MLL datasets, we randomly select \( r \) irrelevant labels and aggregate them with the ground-truth labels to obtain a candidate set. For example, given an instance \( x_i \) and its ground-truth label set \( Y \), we select \( r \) labels from its complementary set \( Y - \tilde{S}_i \). If there are less than \( r \) irrelevant labels, i.e. \( |Y - \tilde{S}_i| < r \), we simply set the whole training set \( Y \) as the candidate set for \( x_i \). Following the experimental settings in [Lyu et al., 2020], we choose \( r \in \{1, 2, 3\} \), resulting in \( 27 \times 9 \) synthetic datasets. Furthermore, we conducted experiments on three real-world PML datasets [Zhang and Fang, 2021], including Music-emotion, Music-style, Mirflickr. These three datasets are derived from the image retrieval task [Huiskes and Lew, 2008], where the candidate labels are collected from web users and then further verified by human annotators to determine the ground-truth labels.

### 4.2 Baselines and Implementation Details

We choose six benchmark methods for comparative studies, including two MLL methods ML-KNN [Zhang and Zhou, 2007], GLOCAL [Zhu et al., 2018], and four state-of-the-art PML methods PML-NI [Xie and Huang, 2022], PARTICLE [Zhang and Fang, 2021], HALE [Lyu et al., 2020], and PML-lc [Xie and Huang, 2018]. In particular, PARTICLE and HALE are two advanced graph-based PML methods. PARTICLE [Zhang and Fang, 2021] adopts an instance-level label propagation algorithm to disambiguate the candidate labels. HALE [Lyu et al., 2020] regards the denoising procedure as a graph-matching procedure.

The parameter setups for the used methods are as follows. We fix \( k = 10 \) for all kNN-based methods, including PLAIN. For the baselines, we fix or fine-tune the hyperparameters as the suggested configurations in respective papers. For our PLAIN method, the deep model is comprised of three fully-connected layers. The hidden sizes are set as \( [64, 64] \) for those datasets with less than 64 labels, and \( [256, 256] \) for those datasets with more than 64 and less than 256 labels. For the remaining datasets, we set the hidden sizes as \( [512, 512] \). The trade-off parameters \( \alpha \) and \( \beta \) are hand-tuned from \( \{0.001, 0.01, 0.1\} \). \( \eta \) is selected from \( \{0.1, 1, 10\} \). Following [Iscen et al., 2017], we set \( \rho = 3 \). We train our deep model via stochastic gradient descent and empirically set the learning rate as 0.01 for both propagation and deep model training procedures. The number of maximum iterations is set as \( T = 200 \) for small-scale datasets and \( T = 50 \) for the large-scale NUS-WIDE dataset. Besides, weight decay is applied with a rate of \( 5e^{-5} \) to avoid overfitting.

For performance measure, we use three widely-used multi-label metrics, ranking loss, average precision and hamming loss [Zhang and Zhou, 2014]. Finally, we perform ten-fold cross-validation on each dataset and the mean metric values as well as the standard deviations are reported.

### 4.3 Experimental Results

Table 1 reports the experimental results on real-world datasets. Table 3 lists the results on synthetic datasets, where the parameter is configured with \( r = 3 \). The results of \( r = 1, 2 \) are consistent with those of \( r = 3 \) and thus are

---

<table>
<thead>
<tr>
<th>Datasets</th>
<th>PLAIN</th>
<th>PML-NI</th>
<th>HALE</th>
<th>ML-KNN</th>
<th>GLOCAL</th>
<th>PML-lc</th>
<th>PARTICLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Music-emotion</td>
<td>0.217±0.006</td>
<td>0.244±0.007</td>
<td>0.299±0.008</td>
<td>0.365±0.010</td>
<td>0.344±0.025</td>
<td>0.310±0.010</td>
<td>0.261±0.007</td>
</tr>
<tr>
<td>Music-style</td>
<td>0.136±0.005</td>
<td>0.138±0.009</td>
<td>0.271±0.015</td>
<td>0.229±0.010</td>
<td>0.253±0.007</td>
<td>0.269±0.016</td>
<td>0.351±0.014</td>
</tr>
<tr>
<td>Mirflickr</td>
<td>0.088±0.006</td>
<td>0.123±0.004</td>
<td>0.143±0.009</td>
<td>0.178±0.013</td>
<td>0.212±0.003</td>
<td>0.206±0.008</td>
<td>0.213±0.011</td>
</tr>
</tbody>
</table>

| Music-emotion     | 0.664±0.007 | 0.610±0.011 | 0.543±0.005 | 0.505±0.010 | 0.513±0.010 | 0.539±0.013 | 0.626±0.011 |
| Music-style       | 0.742±0.012 | 0.740±0.013 | 0.687±0.013 | 0.659±0.014 | 0.645±0.007 | 0.598±0.009 | 0.621±0.016 |
| Mirflickr         | 0.847±0.012 | 0.792±0.004 | 0.718±0.005 | 0.698±0.013 | 0.679±0.007 | 0.675±0.010 | 0.690±0.012 |

Table 1: Comparison of PLAIN with baselines on real-world datasets, where the best results are shown in bold face.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Emotions</th>
<th>Birds</th>
<th>Medical</th>
<th>Image</th>
<th>Corel5k</th>
<th>Bitext</th>
<th>Eurlex-nc</th>
<th>Eurlex-sm</th>
<th>NUS-WIDE</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking Loss</td>
<td>18/0/0</td>
<td>15/0/3</td>
<td>15/0/3</td>
<td>15/1/2</td>
<td>18/0/0</td>
<td>18/0/0</td>
<td>15/0/3</td>
<td>14/4/0</td>
<td>16/0/2</td>
<td>144/5/13</td>
</tr>
<tr>
<td>Average Precision</td>
<td>18/0/0</td>
<td>16/1/1</td>
<td>15/0/3</td>
<td>18/0/0</td>
<td>16/0/2</td>
<td>16/1/1</td>
<td>18/0/0</td>
<td>18/0/0</td>
<td>18/0/0</td>
<td>153/2/7</td>
</tr>
<tr>
<td>Hamming Loss</td>
<td>15/1/2</td>
<td>14/1/3</td>
<td>11/4/3</td>
<td>11/1/6</td>
<td>5/13/0</td>
<td>11/7/0</td>
<td>8/10/0</td>
<td>14/4/0</td>
<td>14/4/0</td>
<td>103/4/14</td>
</tr>
</tbody>
</table>

Table 2: Win/tie/loss counts of PLAIN’s performance against baselines on synthetic data sets (pairwise t-test at 0.05 significance level).
amongst competing algorithms. Based on these results, we
omitted. To better understand the superi- ority of the proposed
method, we summarize the win/tie/loss counts on all syn-
thetic datasets in Table 2. We also report the statistical testing
results in Appendix B for analyzing the relative performance
amongst competing algorithms. Based on these results, we
can draw the following observations:

• The proposed PLAIN method signifi- cantly outperforms
all other competing approaches. For example, on the
Music-emotion and Mirflicker datasets, in terms of aver-
gage precision, PLAIN achieved results superior to those
of the best baselines by 6.07% and 6.94% respectively.

• According to Table 2, out of 486 statistical comparisons,
six competing algorithms, and three evaluation met-
ric s, PLAIN beats the competing methods in 82.30% cases. In particular, PLAIN wins or ties the baselines in
95.68% cases regarding average precision.

• ML-KNN and GLOCAL obtain inferior performance
on many datasets, such as Medical and Eurlex-dc. The
reason is that these tailored MLL models regard all the
candidate labels as the ground-truth and thus tend to overfit
the false-positive labels.

• Two PML methods PML-NI and HALE are strong base-

### Table 3: Comparison of PLAIN with baselines on nine synthetic PML datasets ($r = 3$), where the best results are shown in bold face.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>PLAIN</th>
<th>PML-NI</th>
<th>HALE</th>
<th>ML-KNN</th>
<th>GLOCAL</th>
<th>PML-lc</th>
<th>PARTICLE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ranking Loss (the lower the better)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emotions</td>
<td>0.158±0.024</td>
<td>0.214±0.029</td>
<td>0.235±0.037</td>
<td>0.241±0.026</td>
<td>0.322±0.053</td>
<td>0.459±0.035</td>
<td>0.259±0.019</td>
</tr>
<tr>
<td>Birds</td>
<td>0.205±0.036</td>
<td>0.187±0.035</td>
<td>0.271±0.061</td>
<td>0.304±0.048</td>
<td>0.302±0.041</td>
<td>0.321±0.021</td>
<td>0.301±0.032</td>
</tr>
<tr>
<td>Medical</td>
<td>0.050±0.011</td>
<td>0.039±0.013</td>
<td>0.169±0.25</td>
<td>0.088±0.019</td>
<td>0.068±0.005</td>
<td>0.058±0.012</td>
<td>0.100±0.021</td>
</tr>
<tr>
<td>Image</td>
<td>0.190±0.015</td>
<td>0.289±0.018</td>
<td>0.192±0.015</td>
<td>0.342±0.026</td>
<td>0.264±0.010</td>
<td>0.467±0.025</td>
<td>0.315±0.073</td>
</tr>
<tr>
<td>Corel5K</td>
<td>0.120±0.004</td>
<td>0.205±0.006</td>
<td>0.259±0.111</td>
<td>0.139±0.007</td>
<td>0.164±0.003</td>
<td>0.169±0.013</td>
<td>0.333±0.050</td>
</tr>
<tr>
<td>Bibtext</td>
<td>0.079±0.005</td>
<td>0.126±0.010</td>
<td>0.601±0.009</td>
<td>0.232±0.006</td>
<td>0.131±0.006</td>
<td>0.342±0.005</td>
<td>0.287±0.010</td>
</tr>
<tr>
<td>Eurlex-dc</td>
<td>0.033±0.002</td>
<td>0.030±0.003</td>
<td>0.079±0.002</td>
<td>0.086±0.012</td>
<td>0.150±0.003</td>
<td>0.071±0.003</td>
<td>0.082±0.013</td>
</tr>
<tr>
<td>Eurlex-sm</td>
<td>0.027±0.001</td>
<td>0.028±0.002</td>
<td>0.040±0.002</td>
<td>0.043±0.003</td>
<td>0.071±0.003</td>
<td>0.082±0.013</td>
<td>0.053±0.002</td>
</tr>
<tr>
<td>NUS-WIDE</td>
<td>0.211±0.003</td>
<td>0.221±0.002</td>
<td>0.239±0.012</td>
<td>0.301±0.011</td>
<td>0.311±0.020</td>
<td>-</td>
<td>0.240±0.015</td>
</tr>
</tbody>
</table>

| Hamming Loss (the lower the better) |        |      |        |        |        |        |          |
| Emotions    | 0.812±0.043 | 0.754±0.016 | 0.751±0.035 | 0.741±0.029 | 0.647±0.030 | 0.573±0.025 | 0.745±0.024 |
| Birds       | 0.606±0.037 | 0.580±0.062 | 0.505±0.054 | 0.453±0.052 | 0.390±0.060 | 0.388±0.035 | 0.431±0.051 |
| Medical     | 0.781±0.039 | 0.849±0.035 | 0.769±0.023 | 0.672±0.039 | 0.751±0.009 | 0.713±0.012 | 0.720±0.044 |
| Image       | 0.774±0.020 | 0.668±0.018 | 0.762±0.016 | 0.627±0.022 | 0.692±0.008 | 0.523±0.019 | 0.689±0.096 |
| Corel5K     | 0.319±0.010 | 0.289±0.011 | 0.231±0.008 | 0.251±0.007 | 0.284±0.007 | 0.247±0.012 | 0.135±0.041 |
| Bibtext     | 0.550±0.014 | 0.537±0.012 | 0.353±0.010 | 0.306±0.006 | 0.438±0.011 | 0.283±0.010 | 0.313±0.012 |
| Eurlex-dc   | 0.740±0.008 | 0.700±0.011 | 0.673±0.006 | 0.603±0.012 | 0.278±0.005 | 0.602±0.012 | 0.630±0.016 |
| Eurlex-sm   | 0.802±0.007 | 0.718±0.011 | 0.739±0.013 | 0.761±0.012 | 0.556±0.008 | 0.582±0.013 | 0.695±0.012 |
| NUS-WIDE    | 0.286±0.004 | 0.274±0.003 | 0.230±0.007 | 0.171±0.015 | 0.177±0.024 | -       | 0.206±0.017 |

- means over long time consumption, and thus, the experimental results are not reported.

4.4 Further Analysis

Complexity Analysis. In Table 4, we report the running
time of different training stages of PLAIN, including the
graph building step, label propagation step (one epoch) and
the deep model training step (one epoch), on datasets with
various scales. First of all, we can see that the instance graph
can be fast built in a few microseconds on all datasets, with
the help of Faiss package [Johnson et al., 2017]. Moreover,
by utilizing the sparse property of instance graph and the fast
computation ability of GPUs, the time complexity of label
propagation is at the same magnitude as that of deep model
training. In particular, on those datasets with fewer labels,
such as Music-emotion and Image, the propagation phase is
as fast as the deep model traveling the whole training dataset.
In Figure 2, (a) and (b) show the parameter sensitivities of
PLAIN to $T$ and $\gamma$. We can observe that PLAIN is stable with varying values of $T$ and $\gamma$. Specifically, PLAIN obtains high average precision even with small $T$ and small $\gamma$. But when $T = 0$ (without label propagation), our PLAIN model degenerates to the trivial DNN model and thus, the performance degrades. It should also be noted that when $T$ is too large, the performance of PLAIN slightly drops because the model tends to overfit. In summary, the proposed PLAIN model requires only a few gradient steps in the label propagation phase and can be efficiently trained.

Comparison between Loss Functions. As discussed in section 3.2, mean square error (MSE) is error-prone to the label noise. Hence, we conducted experiments to compare the performance of different loss functions, including MSE, MAE, BCE and probabilistic MSE (PMSE). The results are shown in Figure 2 (c). We can see that MAE demonstrates inferior performance and low convergence. Despite its robustness, it tends to produce harsh predictions and is opposite to the graph techniques which generate smooth outputs. MSE obtains good performance, but underperforms BCE and PMSE, since it easily overfits the false-positive labels. Finally, both BCE and PMSE demonstrate good empirical results. These observations suggest that a robust loss function is essential in successful learning from PML data.

Ablation Analysis. An ablation study would be informative to see how different components of the proposed PLAIN contribute to the prediction performance. Thus, we compare PLAIN with three variants: 1) a pure deep neural network model (DNN) that treats the candidate labels as valid ones; 2) the PLAIN model without considering label-level similarity (PLAIN w/o LS); 3) the PLAIN model without considering instance-level similarity (PLAIN w/o IS). Figure 3 reports the results on one real-world dataset as well as two synthetic datasets. In general, both graph regularizers improve the simple DNN model on most datasets and evaluation metrics. Then, by integrating the two regularizers, our PLAIN model consistently achieves the best performance. These results clearly verify the superiority of the proposed method.

5 Conclusion

In this work, we proposed a novel deep partial multi-label learning model PLAIN that copes with several limitations in existing graph-based PML methods. At each training epoch, we first involve both the instance- and label-level similarities to generate pseudo-labels via label propagation. We then train the deep model to fit the disambiguated labels. Moreover, we analyzed the time complexity and the robustness of PLAIN from both theoretical and empirical perspectives. Comprehensive experiments on synthetic datasets as well as real-world PML datasets clearly validate the efficiency, robustness, and effectiveness of our proposed method. Our work demonstrates that combining the graph and deep models together leads to high expressiveness as well as good disambiguation ability. We hope our work will increase attention toward a broader view of graph-based deep PML methods.
Acknowledgments

This work is supported by the National Key R&D Program of China (No. 2022YFB3304100) and by the Zhejiang University-China Zhejiang Bank Co., Ltd. Joint Research Center. Gengyu Lyu is supported by the National Key R&D Program of China (No. 2022YFB3103104) and the China Postdoctoral Science Foundation (No. 2022M720320). Haobo Wang also wants to thank the support of the NSFC Grants (No. 62206247) and the Fundamental Research Funds for the Central Universities.

References


