

Partial Multilabel Learning Using Noise-Tolerant Broad Learning System With Label Enhancement and Dimensionality Reduction

Wenbin Qian^{1b}, Yanqiang Tu, Jintao Huang, Wenhao Shu, and Yiu-Ming Cheung^{2b}, *Fellow, IEEE*

Abstract—Partial multilabel learning (PML) addresses the issue of noisy supervision, which contains an overcomplete set of candidate labels for each instance with only a valid subset of training data. Using label enhancement techniques, researchers have computed the probability of a label being ground truth. However, enhancing labels in the noisy label space makes it impossible for the existing partial multilabel label enhancement methods to achieve satisfactory results. Besides, few methods simultaneously involve the ambiguity problem, the feature space’s redundancy, and the model’s efficiency in PML. To address these issues, this article presents a novel joint partial multilabel framework using broad learning systems (namely BLS-PML) with three innovative mechanisms: 1) a trustworthy label space is reconstructed through a novel label enhancement method to avoid the bias caused by noisy labels; 2) a low-dimensional feature space is obtained by a confidence-based dimensionality reduction method to reduce the effect of redundancy in the feature space; and 3) a noise-tolerant BLS is proposed by adding a dimensionality reduction layer and a trustworthy label layer to deal with PML problem. We evaluated it on six real-world and seven synthetic datasets, using eight state-of-the-art partial multilabel algorithms as baselines and six evaluation metrics. Out of 144 experimental scenarios, our method significantly outperforms the baselines by about 80%, demonstrating its robustness and effectiveness in handling partial multilabel tasks.

Index Terms—Broad learning system (BLS), dimensionality reduction, granular computing, label enhancement, noisy labels, partial multilabel learning (PML).

Manuscript received 21 July 2023; revised 15 November 2023 and 17 December 2023; accepted 30 December 2023. Date of publication 30 January 2024; date of current version 6 February 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62366019; in part by the National Key Research and Development Program of China under Grant 2022YFD1600202; in part by the Natural Science Foundation of Jiangxi Province, China, under Grant 20224BAB202020; in part by the NSFC/Research Grants Council (RGC) Joint Research Scheme under Grant N_HKBU214/21; in part by the General Research Fund of RGC under Grant 12201321 and Grant 12202622; and in part by the Jiangxi Province Postgraduate Innovation Special Fund Project under Grant YC2022-s390. (*Corresponding authors: Wenbin Qian; Yiu-Ming Cheung.*)

Wenbin Qian is with the School of Software, Jiangxi Agricultural University, Nanchang 330045, China (e-mail: qianwenbin1027@126.com).

Yanqiang Tu is with the School of Computer and Information Engineering, Jiangxi Agricultural University, Nanchang 330045, China (e-mail: tuyanqiang@stu.jxau.edu.cn).

Jintao Huang and Yiu-Ming Cheung are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China (e-mail: csjthuang@comp.hkbu.edu.hk; ymc@comp.hkbu.edu.hk).

Wenhao Shu is with the School of Information Engineering, East China Jiaotong University, Nanchang 330013, China (e-mail: shuwenhao@126.com). This article has supplementary downloadable material available at <https://doi.org/10.1109/TNNLS.2024.3352285>, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2024.3352285

I. INTRODUCTION

PARTIAL multilabel learning (PML) is an entirely new weakly supervised learning paradigm [1], aiming to construct a multilabel model with uncertain data, where each sample is connected with a group of possible labels with only partial correctness. The PML data is quite common in crowdsourcing systems, ecological informatics, and online queries, [2], [3], [4], just to name a few. For example, in crowdsourcing scenario (as illustrated in Fig. 1), suffering from the possible unreliable annotators with varying expertise levels, only a subset of the set of candidate labels is valid.

The main difficulty of learning from PML data is how to diminish the impact of noisy labels. Most existing methods use disambiguation strategies based on label confidence estimation, which are classified into embedded and decoupled methods. Typically, embedded methods integrate the computation process of label confidence and the training process of the learner into one, and both are done in the same optimization process [5], [6], [7]. In contrast, decoupled methods first eliminate some candidate labels using disambiguation and then use the remaining candidate labels to induce the desired models on off-the-shelf multilabel learning (MLL) models [8], [9], [10]. Among them, label enhancement techniques get benefit from topology of the data to help recover the confidence of the labels, thus making them effectively to estimate label confidence [10], [11], [12]. The existing label enhancement methods fail to consider the sparsity of data and the ambiguity of labels simultaneously. This leads to the enhancement results still being affected by noisy labels, thus degrading the model’s performance. Another challenge of PML comes from the ambiguity and redundancy of the feature space [13], [14], [15], which are often cited as one of the causes of noisy labels [16], [17], [18]. However, existing methods consider only one of ambiguity and redundancy but not both. From a practical perspective, both of them may be present simultaneously. Finally, existing PML learning models require multiple iterations to minimize the loss function, which cannot balance efficiency and accuracy.

The broad learning system (BLS) [19] is a novel and efficient neural network structure that has been applied to various data scenarios in recent years. For example, Huang et al. [13] designed a BLS for large-scale multilabel data by adding a feature selection layer, and Liu et al. [20] proposed a BLS with an adaptive reweighting strategy to deal with the



Fig. 1. In a PML scenario, during image annotation, three out of the four candidate labels are valid (shown in black).

problem of noisy labeling in the single-label scenario. The main advantages of BLS are that it can build a broad network architecture with sparse autoencoders and enhancement nodes, reduce the output weights optimization to a least square problem that can be solved quickly by pseudoinverse, and allow incremental learning of new samples or nodes without retraining the whole network.

Obviously, BLS may achieve superior performances when tackling PML issues benefiting from its joint and efficient learning capability. Nevertheless, suffering from the combination of multilabel, feature redundancy, and noise labeling, the existing BLS model and its variants are unable to directly and effectively unravel the PML problems. To extend the BLS to handle these problems, this article proposes a novel noise-tolerant BLS structure by adding two novel layers: at the trustworthy label layer, the label enhancement is performed in the trusted label space to avoid the interference of noisy labels. Meanwhile, introducing neighborhood can effectively alleviate the data sparsity problem. At the dimensionality reduction layer, this article proposes a kernel linear discriminant analysis method based on label confidence prior probabilities, which can project the original space to a higher dimensional space and achieve dimensional reduction, thus enabling the model to mitigate the impact of feature ambiguity while achieving the goal of reducing feature redundancy. The model can remain efficient and resist noise by reconstructing the BLS. The contributions in our article can be summarized as follows.

- 1) A label enhancement method for partial multilabel data is proposed: the label enhancement is performed in the ideal label space generated based on matrix factorization to avoid the bias caused by the label enhancement in the original space. The objective function is constructed by local sample similarity and global label correlation constraints to recover the label confidence for each training instance accurately.
- 2) A confidence-based kernel linear discriminant analysis method for dimensionality reduction is proposed: to alleviate the effect of noisy labels, the prior class probability was calculated according to the confidence matrix. Then, a linear discriminant analysis method was designed to reduce the feature dimension, thereby reducing the redundancy of the feature space and further preventing the potential noise in the features.
- 3) A noise-tolerant BLS for predict model is proposed: reconstructing the original BLS, by adding a kernel dimensionality reduction layer and a trustworthy label layer, a novel BLS is constructed, which can deal with the partial multilabel problem and maintain efficiency.

The structure of the remaining content is as follows. First, in Section II, a brief overview of related works is provided. Following that, Section III presents the technical details of the BLS-PML approach. Subsequently, the extensive experimental results are presented in Section IV. Finally, a summary of the proposed approach is provided.

II. RELATED WORK

In this section, we provide a summary of the pertinent research on PML and BLS, both of which are closely connected to the proposed approach.

A. Partial Multilabel Learning

PML framework presents unique challenges that researchers have been actively solving in recent years. One simple approach to handling PML modeling is to treat all candidate labels as true and use existing MLL algorithms to produce the desired multilabel predictive model. However, it is clear that the existence of noisy labels in the candidate label set can impede the success of this easy strategy. Therefore, some learning methods have been specifically developed for PML problems. In general, ready-made PML approaches can be classified into embedded and decoupled methods.

For example, some of the methods that employ embedded approaches include: Sun et al. [21] handled the PML problem by leveraging label correlation information. Yan et al. [22] tackled noisy labels through the adoption of a matching teacher network and a prediction network. Cao et al. [23] distinguished the noisy labels from the candidate label set by optimizing the distribution of ranking margin. Li et al. [24] proposed a method to reduce noisy labels and redundant features by learning a latent label subspace and a feature subspace.

In contrast, other methods use decoupled approaches. Wang et al. [8] utilized the feature manifold to learn a confidence value first. Then, a gradient boosting model was used to fit the confidence. Zhang and Fang [25] used iterative label propagation to get the confidence of candidate labels first. Then, they built a predictor with high-confidence labels. He et al. [9] first used a soft threshold operator to enlarge the confidence difference of labels, and then, the model learned in the relabel space.

The methods above are based on the calculation of the label's confidence. They often ignore the effective use of sample similarity and label correlation, so we use the neighborhood granularity to explore the local sample similarity to prevent the influence of sparse samples on the model, and jointly use the sample similarity and label correlation in the clean label space for label enhancement, so as to recover the confidence of the ground-truth label.

B. Broad Learning System

The BLS is a shallow neural network, which was first proposed by Chen and Liu [19], building on the random vector functional link neural network (RVFLNN) [26], [27], [28]. BLS comprises feature nodes, enhancement nodes, and output coefficients and has a simple structure that enables effective

feature extraction and maintenance of data validity. The output coefficients connect each node with the objective matrix, and their values are obtained by pseudoinverse to ensure high modeling efficiency without the need for iterative updating such as deep neural networks.

Thanks to the effectiveness and efficiency of BLS, researchers have solved many problems based on BLS [29], [30], [31]. Jin et al. [32] proposed a robust BLS model that can handle noisy labels in single-label scenario by using maximum likelihood estimation and manifold regularization. Yu et al. [33] used kernel BLS to reduce uncertainty and tuning, and progressive ensemble to enhance stability and noise resistance. Li et al. [34] proposed a regularized BLS for face sketch synthesis, which can directly transform photographs into sketches with rich details and low computational complexity. Han et al. [35] put forward the concept of the ‘‘Maximum Information Exploitation BLS’’ to address the challenge of extreme information utilization in modeling large-scale chaotic time series. Jin and Chen [36] proposed an RBLS model that can handle outliers in data by using maximum a posterior estimation and regularization theory. Zheng et al. [37] used the maximum correntropy criterion to train the output weights and enhance the robustness of BLS to outliers. Chu et al. [38] used a weighted penalty factor to constrain the contribution of each sample to improve the model performance.

Despite the success of these different models in their respective scenarios, none can be directly applied to the PML problem. Therefore, to fill this research gap and deal with the PML problem more efficiently, this article modifies the input layer and label layer of the original BLS and designs a noise-tolerant BLS for partial multilabel data.

III. PROPOSED METHOD

The proposed BLS-PML method contains two main mechanisms: a matrix decomposition-based label enhancement module and a confidence-based K-LDA dimensionality reduction module. In the label enhancement module, the similarity of the samples and the correlation of the labels are used to perform label enhancement in the trustworthy label space. This space is generated by the matrix decomposition based on the low-rank sparsity assumption. The information from the label enhancement is used to help recover the confidence of the labels. In the dimensional reduction module, the original feature space is mapped to higher dimensions to increase representational power, where dimensional reduction is performed using linear discriminant analysis based on label confidence priors to reduce feature ambiguity and redundancy. Finally, the obtained label confidence and the reduced features are used together to construct a BLS for partial multilabel data. The configuration of BLS-PML is depicted in Fig. 2. The technical details are presented in this section.

A. Notations

In the PML dataset, let $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{n \times d}$ denote a feature space with n sample d -dimensional features. Correspondingly, $Y = [y_1, y_2, \dots, y_k]^T \in \{0, 1\}^{k \times n}$ denote the candidate label space of n sample k -dimensional labels.

The presence of noisy labels makes labels with a value of 1 in the set of candidate labels less than fully plausible, while labels with a value of 0 are determined to be irrelevant.

B. Partial Multilabel Label Enhancement

In PML, the ground-truth label is concealed within the candidate label set. The primary goal of this method is to retrieve the ground-truth label from the candidate label set, so the candidate label set is split into a clean label matrix C and a noise label matrix N . Then, choose the least square loss to construct the objective function

$$\min_{C, N} \frac{1}{2} \|Y - C - N\|_F^2. \quad (1)$$

In general, the clean label matrix C where element $c_{ij} \geq 0$ represents the probability that the i th sample being associated with the j th label, which is the label confidence. In addition, past studies have shown that using label correlation to solve multilabel problems is an effective method. Therefore, there is correlation between labels in the clean label space, which makes the clean label matrix has low-rank property. Thus, (1) can be rewritten as

$$\min_{C, N} \frac{1}{2} \|Y - C - N\|_F^2 + \text{rank}(C) \quad (2)$$

where $\text{rank}(C)$ is the rank of matrix C , but minimizing the rank of a matrix is NP-hard. According to the sparse optimization, $\text{rank}(C)$ is replaced by finding the sum of all singular values, which is the nuclear norm of matrix C , i.e., $\|C\|_* = \sum_i \sigma_i(C)$

In addition, compared with the huge label space, the candidate label space is relatively sparse, so the noise labels as a subset of candidate labels should be more sparse, and the labelers have certain cognitive ability, and the quality of the labels given by them should not be too poor. Based on the above discussion, the ℓ_0 norm is used to constrain the sparsity of the noise matrix N , so (2) is rewritten as

$$\min_{C, N} \frac{1}{2} \|Y - C - N\|_F^2 + \|C\|_* + \lambda_1 \|N\|_0 \quad (3)$$

where λ_1 is the hyperparameter that balances the proportion of the term in the objective function. When $\lambda_1 = 0$, the problem degenerates into a general multilabel problem. $\|N\|_0$ refers to the number of non-zero elements in N . Since $\|N\|_0$ is a discontinuous function, its value can only be an integer, and its solution is also an NP-hard problem, which is replaced by ℓ_1 norm here, i.e., $\|N\|_1 = \sum_{i=1}^n |N|$, so (3) is rewritten into as

$$\min_{C, N} \frac{1}{2} \|Y - C - N\|_F^2 + \|C\|_* + \lambda_1 \|N\|_1. \quad (4)$$

To fully exploit the inherent relationships within the provided data, the study of feature similarity and label correlation will be focused here. First, based on the smoothness assumption, samples with similar feature should be similar in the label space as much as possible. Different from previous studies that explore feature similarity in the global space or k -nearest neighbor space, we use the neighborhood to determine the

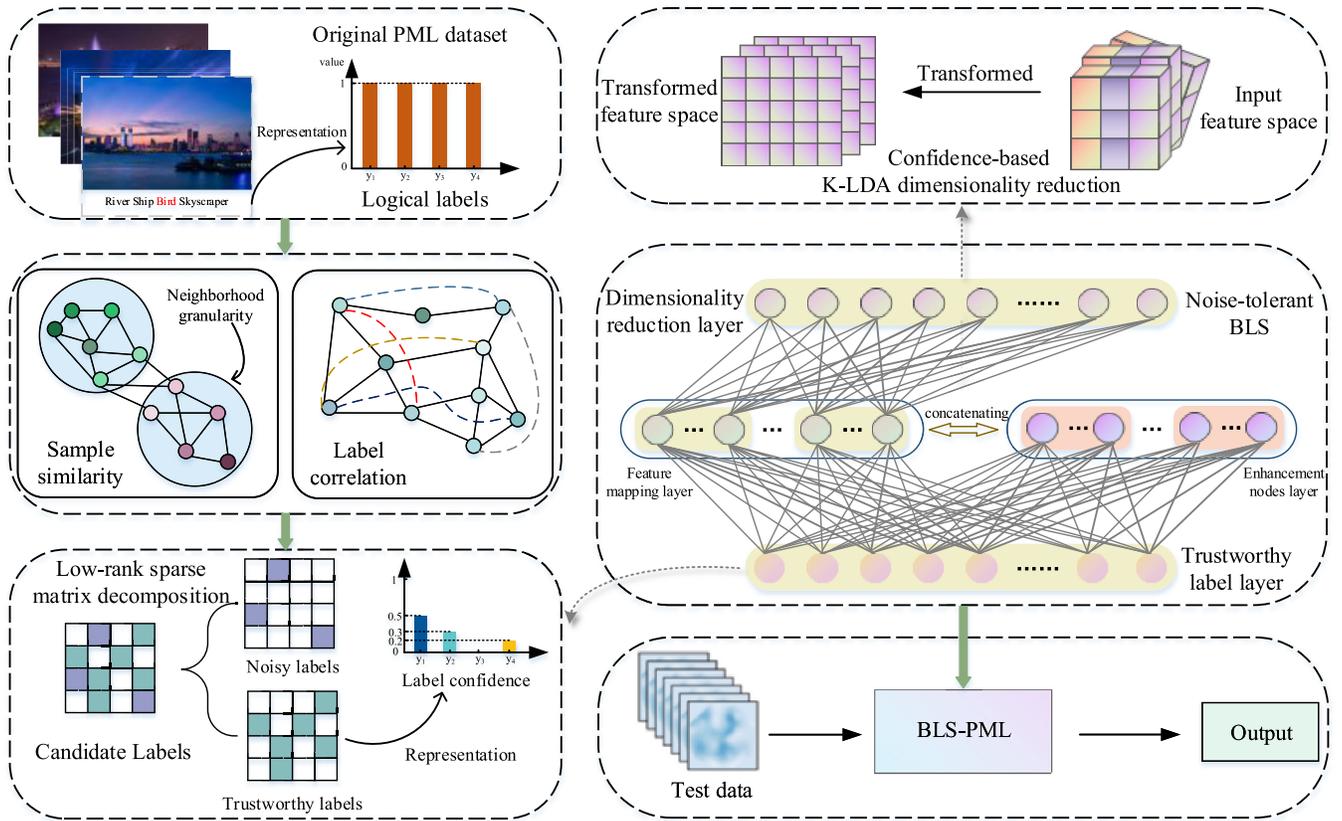


Fig. 2. Structure of BLS for PML (BLS-PML). The three dashed boxes located on the left are modifications to the label layer and are a brief illustration of Section III-B. The first dashed box on the right is a transformation of the input layer and corresponds to Section III-C. The dashed box in the lowermost right corner is a prediction process for the unseen example.

similar sample set for each sample. This method has the advantages of capturing more accurate local sample relationships and avoiding the issue of outliers that a fixed k value cannot handle. Specifically, the similarity between samples is obtained by a Gaussian function in the locality of the neighborhood granularity of sample x , and the neighborhood granularity of samples in the feature space can be expressed as

$$\delta_{x_i} = \{x | x \in X, \Delta(x, x_i) \leq \delta\} \quad (5)$$

where Δ represents the distance between two samples and δ is the threshold of neighborhood granularity. Based on the neighborhood granularity space, the similarity matrix O can be defined as

$$o_{ij} = \begin{cases} \exp(-\|x_i - x_j\|_2^2 / 2\sigma^2), & \text{if } x_i \in \delta_{x_j} \text{ or } x_j \in \delta_{x_i} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where $\sigma > 0$ is a user-specified Gaussian function bandwidth parameter. After that, a constrained regularization term based on sample similarity is constructed according to the smoothness assumption

$$\min_C \sum_{i=1}^n \sum_{j=1}^n \|c_i - c_j\|^2 o_{ij} = \text{tr}(C^T L C) \quad (7)$$

where $L = D - O$ is a Laplacian matrix and D is a diagonal matrix whose diagonal elements are the sum of the row elements of the underlying matrix O , i.e., $D_{ii} = \sum_{j=1}^n o_{ij}$, O matrix encodes the similarity between sample features.

In addition, it is necessary to consider the correlation between the labels for the recovery of true labels. For example, white clouds and sky often appear together, so when white cloud appears in a label, the probability of sky being a noisy label is small. Based on this assumption, the Pearson correlation coefficient is used to measure the label correlation in the global label space. The reason why the global label is not local is that the local is more susceptible to the interference of noisy labels than the global label. The label correlation coefficient can be defined as

$$r_{ij} = \begin{cases} 1, & P(y_{:,i}, y_{:,j}) > 0 \\ 0, & P(y_{:,i}, y_{:,j}) = 0 \\ -1, & P(y_{:,i}, y_{:,j}) < 0 \end{cases} \quad (8)$$

where $P(y_{:,i}, y_{:,j})$ denotes the Pearson correlation coefficient that measures between two labels

$$P(y_{:,i}, y_{:,j}) = \frac{\sum_{k=1}^n (y_{ki} - \bar{y}_{:,i})(y_{kj} - \bar{y}_{:,j})}{\sqrt{\sum_{k=1}^n (y_{ki} - \bar{y}_{:,i})^2 (y_{kj} - \bar{y}_{:,j})^2}} \quad (9)$$

As mentioned above, higher correlation of different labels can achieve closer confidence levels presented in the process of ground-truth label recovery. Based on this assumption, a constraint regularization term based on label correlation is constructed to further restrict the objective function

$$\min_C \sum_{i=1}^k \sum_{j=1}^k r_{ij} \|c^i - c^j\|^2 = \text{tr}(C S C^T) \quad (10)$$

where c^i denotes the confidence vector constructed by all samples on the i th class label, $S = G - R$ is a Laplacian matrix, and G is a diagonal matrix whose diagonal elements are the sum of the row elements of the underlying matrix R , i.e., $G_{ii} = \sum_{j=1}^k r_{ij}$. The correlation between labels is encoded in the R matrix.

Combining (7) and (10), and incorporating feature similarity and label correlation into the objective function, (4) can be rewritten as

$$\min_{C, N} \frac{1}{2} \|Y - C - N\|_F^2 + \|C\|_* + \lambda_1 \|N\|_1 + \lambda_2 [\text{tr}(C^T LC) + \text{tr}(CSC^T)] \quad (11)$$

where λ_2 is the hyperparameter that balances the proportion of the term in the objective function.

1) *Optimization*: As shown in (11), the objective function contains the matrix nuclear norm and ℓ_1 norm regularization term, so (11) is a non-smooth convex function, and the accelerated proximal gradient (APG) algorithm [39] is used to solve the problem. For the two unknown variables C and N in (11), the alternating optimization method is used to deal with them. The detailed process is as follows.

According to the APG algorithm, $\min_Z H(Z) = f(Z) + g(Z)$, (11) can be split into differentiable and non-differentiable convex functions, that is,

$$\begin{aligned} g(Z) &= \|C\|_* + \lambda_1 \|N\|_1 \\ f(Z) &= \frac{1}{2} \|Y - C - N\|_F^2 + \frac{\lambda_2}{2} [\text{tr}(C^T LC) + \text{tr}(CSC^T)]. \end{aligned} \quad (12)$$

Among them, $Z = \begin{pmatrix} C \\ N \end{pmatrix}$, $f(Z)$ is a continuously differentiable convex function and the gradient is Lipschitz continuous, that is, $\|\nabla f(Z_1) - \nabla f(Z_2)\| \leq L_f \|Z_1 - Z_2\|$, and L_f is the Lipschitz constant. For $f(Z)$, we can compute $\nabla f(Z)$ as follows:

$$\begin{aligned} \nabla f(C) &= \lambda_2(LC + CS) + C + N - Y \\ \nabla f(N) &= C + N - Y. \end{aligned} \quad (13)$$

Let $Z_1 = \begin{pmatrix} C_1 \\ N_1 \end{pmatrix}$ and $Z_2 = \begin{pmatrix} C_2 \\ N_2 \end{pmatrix}$. Then, according to the Lipschitz condition, we can obtain

$$\begin{aligned} &\|\nabla f(Z_1) - \nabla f(Z_2)\|_F^2 \\ &\leq [4\sigma_{\max}^2(\lambda_2 L) + 4\sigma_{\max}^2(\lambda_2 S) + 6] \|\Delta C\|_F^2 + 6\|\Delta N\|_F^2 \\ &\leq L_f^2 \left\| \begin{pmatrix} \Delta C \\ \Delta N \end{pmatrix} \right\|_F^2 \end{aligned} \quad (14)$$

where $\Delta C = C_1 - C_2$ and $\Delta N = N_1 - N_2$, $\sigma_{\max}(\cdot)$ returns the maximum singular value in a matrix, so Lipschitz constant can be computed as follows:

$$L_f = \sqrt{4\sigma_{\max}^2(\lambda_2 L) + 4\sigma_{\max}^2(\lambda_2 S) + 6}. \quad (15)$$

- 1) *Updating C*: To solve the objective function using the accelerated proximal gradient algorithm, instead of minimizing $H(Z)$ directly, the algorithm minimizes a series of separable quadratic approximations of $H(Z)$, denoted $Q(Z, M)$, formed at a specially chosen point M .

$$\begin{aligned} Q(Z, M) &\approx f(M) + \langle \nabla f(M), Z - M \rangle \\ &\quad + \frac{L_f}{2} \|Z - M\|^2 + g(Z). \end{aligned} \quad (16)$$

Let k be the k th iteration, we can obtain

$$\begin{aligned} Z_{k+1} &= \arg \min_Z Q(Z, M) \\ &= \arg \min_Z f(M_k) + \langle \nabla f(M_k), Z - M_k \rangle \\ &\quad + \frac{L_f}{2} \|Z - M_k\|^2 + g(Z) \end{aligned} \quad (17)$$

where $Z = \begin{pmatrix} C \\ N \end{pmatrix}$ and $M = \begin{pmatrix} M_k^C \\ M_k^N \end{pmatrix}$. When updating C_{k+1} , N is set to N_k and treated as a constant. Thus, C_{k+1} can be calculated by

$$\begin{aligned} C_{k+1} &= \arg \min_C f(M_k) + \frac{L_f}{2} \|C - M_k^C\|_F^2 + g(C) \\ &\quad + \langle \lambda_2(LM_k^C + M_k^C S) + M_k^C + M_k^N - Y, C - M_k^C \rangle. \end{aligned} \quad (18)$$

Let $O_k^C = \lambda_2(LM_k^C + M_k^C S) + M_k^C + M_k^N - Y$, Then, (18) can be rewritten as

$$\begin{aligned} C_{k+1} &= \arg \min_C \frac{L_f}{2} \left\| C - M_k^C + \frac{1}{L_f} O_k^C \right\|_F^2 + \|C\|_* \\ &\quad + \lambda_1 \|N_k\|_1 + f(M_k) - \frac{1}{2L_f} \|O_k^C\|_F^2. \end{aligned} \quad (19)$$

In the process of updating C , $\lambda_1 \|N_k\|_1 + f(M_k) - 1/(2L_f) \|O_k^C\|_F^2$ is treated as a constant, so (19) can be rewritten as

$$C_{k+1} = \arg \min_C \frac{1}{L_f} \|C\|_* + \frac{1}{2} \left\| C - \left(M_k^C + \frac{1}{L_f} O_k^C \right) \right\|_F^2. \quad (20)$$

Let $G_k^C = M_k^C + 1/(L_f) O_k^C$, to speed up convergence, set $M_k^C = C_k + (b_{k-1} - 1)/b_k (C_k - C_{k-1})$, of which b_k satisfies $b_{k+1}^2 - b_{k+1} \leq b_k^2$, and then (20) can be rewritten as

$$C_{k+1} = \arg \min_C \frac{1}{L_f} \|C\|_* + \frac{1}{2} \|C - G_k^C\|_F^2. \quad (21)$$

Soft-thresholding operators were introduced to solve (21) with definition as [40]

$$\mathcal{S}_\varepsilon[z] \approx \begin{cases} z - \varepsilon, & \text{if } z > \varepsilon \\ z + \varepsilon, & \text{if } z < -\varepsilon \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

where $\varepsilon > 0$, it represents the step size of the proximal gradient descent. Then, C_{k+1} is given by soft-thresholding the entries of G_k as

$$C_{k+1} = \arg \min_C \varepsilon \|C\|_* + \frac{1}{2} \|C - G_k^C\|_F^2 = U \mathcal{S}_\varepsilon[\Sigma] V^T \quad (23)$$

where $\varepsilon = 1/L_f$, and $U \Sigma V^T$ is the singular value decomposition (SVD) of G_k , i.e., $G_k = U \Sigma V^T$. In addition, $\mathcal{S}_\varepsilon[\Sigma]$ is a diagonal matrix and its diagonal elements are computed as $(\mathcal{S}_\varepsilon[\Sigma])_{ii} = \max\{0, \Sigma_{ii} - 1/L_f\}$.

- 2) *Updating N*: In the process of updating N , C is fixed to a constant C_{k+1} , and then N_{k+1} can be computed.

$$N_{k+1} = \arg \min_N f(M_k) + \frac{L_f}{2} \|N - M_k^N\|_F^2 + g(N) + \langle M_k^C + M_k^N - Y, N - M_k^N \rangle. \quad (24)$$

Let $O_k^N = M_k^C + M_k^N - Y$, then (24) can be rewritten as

$$N_{k+1} = \arg \min_N \|C_{k+1}\|_* + \lambda_1 \|N\|_1 + \frac{L_f}{2} \left\| N - M_k^N + \frac{1}{L_f} O_k^N \right\|_F^2 + f(M_k) - \frac{1}{2L_f} \|O_k^N\|_F^2 \quad (25)$$

where $\|C_{k+1}\|_* + f(M_k) - 1/(2L_f)\|O_k^N\|_F^2$ is treated as a constant, so (25) can be rewritten as

$$N_{k+1} = \arg \min_N \frac{\lambda_1}{L_f} \|N\|_1 + \frac{1}{2} \left\| N - \left(M_k^N - \frac{1}{L_f} O_k^N \right) \right\|_F^2. \quad (26)$$

Let $G_k^N = M_k^N - (1/L_f)O_k^N$, here set $M_k^N = N_k + (b_{k-1} - 1)/b_k(N_k - N_{k-1})$. According to [40], N_{k+1} can be calculated by

$$N_{k+1} = S_{\frac{\lambda_1}{L_f}}[G_k^N]. \quad (27)$$

Algorithm 1 Partial Multilabel Label Enhancement

Input: Partial multi-label data: $X \in \mathbb{R}^{n \times d}$ and $Y \in \{0, 1\}^{k \times n}$, and weighting parameters: λ_1, λ_2

Output: Trusted label matrix: C

- 1 Initialize the number of iterations: $k \leftarrow 1$; sequence: $b_0, b_1 \leftarrow 1$;
 - 2 **repeat**
 - 3 Calculate Lipschitz constant L_f by (15);
 - 4 $M_k^C = C_k + \frac{b_{k-1}-1}{b_k}(C_k - C_{k-1})$;
 - 5 $G_k^C = M_k^C + \frac{1}{L_f}(\lambda_2(LM_k^C + M_k^C S) + M_k^C + M_k^N - Y)$;
 - 6 Calculate C_{k+1} according to 23;
 - 7 $M_k^N = N_k + \frac{b_{k-1}-1}{b_k}(N_k - N_{k-1})$;
 - 8 $G_k^N = M_k^N - \frac{1}{L_f}(M_k^C + M_k^N - Y)$;
 - 9 Calculate N_{k+1} according to 27;
 - 10 $k \leftarrow k + 1$;
 - 11 **until** stop criterion is reached;
 - 12 $C \leftarrow C_k$
-

Algorithm 1 shows the whole process of partial multilabel label enhancement. Step 1 is to initialize, to calculate and solve the trustworthy label matrix. Steps 2–10 are an alternating optimization process, where step 3 is to calculate the Lipschitz constant, step 6 is to obtain the clean label matrix of iteration $k + 1$, step 9 is to obtain the noisy label matrix of iteration $k + 1$, until the stopping condition is reached, and finally a clean label matrix in the form of label confidence is obtained.

C. Confidence-Based K-LDA Dimensionality Reduction

To reduce the hidden noise in the features, the kernel-based linear discriminant analysis is used to reduce the dimension of the original feature space.

Classical multiclass linear discriminant analysis obtains the projection matrix W by

$$\arg \max_W \text{tr}(W^T S_b W) \quad \text{s.t.}: w_h^T S_w w_h = 1, \quad (1 \leq h \leq p) \quad (28)$$

where $S_b \in \mathbb{R}^{d \times d}$ represents the between-class scatter matrix and $S_w \in \mathbb{R}^{d \times d}$ represents the within-class scatter matrix. p denotes the dimension after the reduction.

To improve the ability to cope with nonlinear data, kernel-based nonlinear discriminant analysis is used for dimensionality reduction. Let $\phi: \mathbb{R}^d \mapsto \mathbb{R}^{\mathcal{H}}$ be the mapping from the original feature space to the Reproducing Kernel Hilbert Space induced by kernel function $\kappa: \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. Then the sample mean of class i before projection and the mean of all samples before projection can be calculated as in (29) and (30), respectively,

$$m_i = \frac{1}{n_i} \sum_{l=1}^{n_i} \phi(x_l^i) = \frac{1}{n_i} X_i^T \mathbf{1}_{n_i \times 1} \quad (29)$$

$$m = \frac{1}{n} \sum_{i=1}^k \sum_{l=1}^{n_i} \phi(x_l^i) = \frac{1}{n} X^T \mathbf{1}_{n \times 1}. \quad (30)$$

It is worth noting that there is no need to consider the noisy labels for the classical linear discriminant analysis. To make classical LDA suitable for PML, the class prior probability is calculated as the ratio of the sum of the label confidence to the total number of samples, i.e., $P_i = \|c_i\|_1/n$, where c_i represents the vector of the confidence of all samples in the training set on class i ($1 \leq i \leq k$). Therefore, the kernelized between-class scattering matrix S_b and within-class scattering matrix S_w can lead to the following results:

$$S_b = \sum_{i=1}^k P_i (m_i - m)(m_i - m)^T \quad (31)$$

$$S_w = \sum_{i=1}^k P_i \sum_{l=1}^{n_i} (x_l^i - m_i)(x_l^i - m_i)^T. \quad (32)$$

Since the function $\phi(\cdot)$ is unknown, (31) and (32) cannot be directly calculated. Here, (31) and (32) will be split and calculated, respectively,

$$S_b = \sum_{i=1}^k P_i (m_i - m)(m_i - m)^T = X^T B X \quad (33)$$

where $\sum_{i=1}^k P_i = 1$, and $B = (1/n) \text{diag}((1/n_1)\mathbf{1}_{n_1 \times n_1}, (1/n_2)\mathbf{1}_{n_2 \times n_2}, \dots, (1/n_k)\mathbf{1}_{n_k \times n_k}) - (1/n^2)\mathbf{1}_{n \times n}$

$$S_w = X^T D X \quad (34)$$

where $D = \text{diag}(P_1 I_{n_1}, P_2 I_{n_2}, \dots, P_k I_{n_k}) - \text{diag}((1/n_1)\mathbf{1}_{n_1 \times n_1}, (1/n_2)\mathbf{1}_{n_2 \times n_2}, \dots, (1/n_k)\mathbf{1}_{n_k \times n_k})$.

Equation (28) can be transformed into (35) by the Lagrange multiplier method

$$S_b W = \lambda S_w W \quad (35)$$

where λ is the Lagrangian multipliers. By substituting (33) and (34) into (35), (35) can be rewritten as follows:

$$X^T B X W = \lambda X^T D X W. \quad (36)$$

Since X is unknown, (36) cannot be solved, so (36) is rewritten as follows:

$$X^T B X X^T \mu = \alpha X^T D X X^T \mu. \quad (37)$$

Since $B X X^T \mu = \alpha D X X^T \mu$, the eigenvalues α and the eigenvector μ can be obtained by solving $(DK)^{-1} B K \mu = \alpha \mu$, where $X X^T = K$. Compared with (36), where $\alpha = \lambda$, $X^T \mu$ is unitized since $X^T \mu$ only represents the projection direction

$$W = \frac{X^T \mu}{\|X^T \mu\|} = \frac{X^T \mu}{\sqrt{\mu^T K \mu}}. \quad (38)$$

The top p , ($p > k - 1$), dimensions are selected to form the projection matrix

$$\begin{aligned} A_p &= [w_1, w_2, \dots, w_p] \\ &= \left[\frac{X^T \mu_1}{\sqrt{\mu_1^T K \mu_1}}, \frac{X^T \mu_2}{\sqrt{\mu_2^T K \mu_2}}, \dots, \frac{X^T \mu_p}{\sqrt{\mu_p^T K \mu_p}} \right] \\ &= X^T \left[\frac{\mu_1}{\sqrt{\mu_1^T K \mu_1}}, \frac{\mu_2}{\sqrt{\mu_2^T K \mu_2}}, \dots, \frac{\mu_p}{\sqrt{\mu_p^T K \mu_p}} \right] \\ &= X^T U_p. \end{aligned} \quad (39)$$

The final dimensionality reduction feature X' can be calculated by

$$X' = A_p^T X = (U_p^T X^T) X = U_p^T K. \quad (40)$$

Algorithm 2 Confidence-Based K-LDA Dimensionality Reduction

Input: Partial multi-label data matrix: $X \in \mathbb{R}^{n \times d}$ and trusted label matrix: $C \in [0, 1]^{k \times n}$, and dimension of the induced feature space: p

Output: The feature space after dimensionality reduction: X'

- 1 Calculate kernel matrix $K = X X^T$ with the specified kernel function $k(\cdot, \cdot)$;
 - 2 Compute class prior probabilities P_i based on label confidence c_i ;
 - 3 Set $B = \frac{1}{n} \text{diag}(\frac{1}{n_1} 1_{n_1 \times n_1}, \frac{1}{n_2} 1_{n_2 \times n_2}, \dots, \frac{1}{n_k} 1_{n_k \times n_k}) - \frac{1}{n^2} 1_{n \times n}$ according to (33);
 - 4 Set $D = \text{diag}(P_1 I_{n_1}, P_2 I_{n_2}, \dots, P_k I_{n_k}) - \text{diag}(\frac{1}{n_1} 1_{n_1 \times n_1}, \frac{1}{n_2} 1_{n_2 \times n_2}, \dots, \frac{1}{n_k} 1_{n_k \times n_k})$ according to 34;
 - 5 Calculate eigenvalues α and eigenvectors μ by $(DK)^{-1} B K \mu = \alpha \mu$;
 - 6 Derive projection matrix A_p according to 39;
 - 7 The first p -dimensional feature matrix X' are induced according to 40;
-

Algorithm 2 summarizes the process of feature reduction. As shown in step 1, the kernel matrix of the feature is calculated by the specified kernel function. Step 2 is to calculate

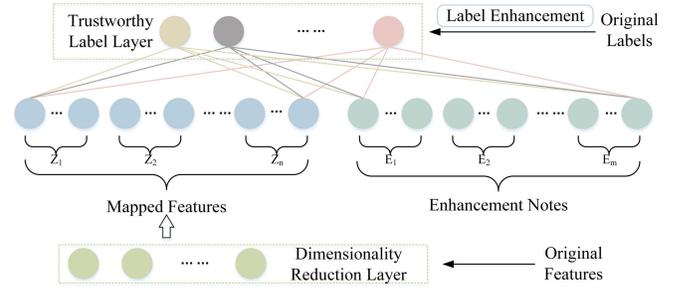


Fig. 3. Architecture of the PML-based noise-tolerant BLS.

the class prior probability according to the label confidence obtained in Algorithm 1. After the calculation of steps 3–7, the required first p -dimensional feature matrix is obtained.

D. PML-Based Noise-Tolerant BLS

The BLS is highly dependent on the accuracy of supervised information in the training process, because it has a shallow number of layers and directly applies pseudoinverse instead of iterative solution. In order to make the BLS be used to process PML data while maintaining its excellent structure, in this section, a noise-tolerant BLS is constructed by adding a dimensionality reduction layer and a trustworthy label layer, and its structure is shown in Fig. 3.

As the figure above shows, after the true label confidence recovery and kernel-based dimension reduction of the original partial multilabel dataset, the original feature space is changed from $X \in \mathbb{R}^{n \times d}$ to $X' \in \mathbb{R}^{n \times p}$, and the original label space is changed from $Y \in \{0, 1\}^{k \times n}$ to $C \in [0, 1]^{k \times n}$. Hence, the feature mapped nodes in the linear transformation of the input data $X' \in \mathbb{R}^{n \times p}$ are defined as follows for n groups with q feature mapped nodes in the feature mapped layer:

$$Z_i = \varphi(X' \theta_{ei} + b_{ei}) \quad (41)$$

where Z_i denotes the i th ($1 \leq i \leq n$) feature mapped node of the feature mapped layer. θ_{ei} and b_{ei} are weights and bias coefficients, which are randomly generated to better map the features. Furthermore, $\varphi(\cdot)$ denotes a linear transformation. After that, all feature mapped nodes are connected, i.e., $Z^n = [Z_1, Z_2, \dots, Z_n]$. Assuming that there are m groups of enhanced nodes in the enhanced nodes layer, the nonlinear transformation of Z^n can be expressed as

$$E_j = \xi([Z_1, Z_2, \dots, Z_n] \theta_{hj} + b_{hj}) \quad (42)$$

where E_j denotes the j th enhancement nodes in each group. Similar to (41), the weight coefficients θ_{hj} and bias coefficients b_{hj} are also randomly generated, respectively. Let $\xi(\cdot)$ be an activation function. Then all enhancement nodes can also be merged as $E^m = [E_1, E_2, \dots, E_m]$. Accordingly, the broad learning network can be expressed as the following equation:

$$\begin{aligned} C &= [Z_1, \dots, Z_n | \xi(Z^n \theta_{h1} + \beta_{h1}), \dots, \xi(Z^n W_{hm} + \beta_{hm})] \theta^m \\ &= [Z_1, \dots, Z_n | E_1, \dots, E_m] \theta^m \\ &= [Z^n | E^m] \theta^m \\ &= A \theta^m. \end{aligned} \quad (43)$$

In (43), θ^m needs to be computed to be able to make predictions on unseen examples. It can be obtained by taking the pseudoinverse of A , i.e., $\theta^m = A^+C$. In BLS, the pseudoinverse of the matrix A is approximated by ridge regression

$$\arg \min_W (\|C - I\theta\|_2^2 + \lambda\|\theta\|_2^2) \quad (44)$$

where λ is the hyperparameter of ℓ_2 regularization. Using the gradient descent algorithm, (45) can be obtained as follows:

$$\theta = (\lambda I + A^T A)^{-1} A^T C \quad (45)$$

where I is the identity matrix. When $\lambda = 0$, $A^+ = \lim_{\lambda \rightarrow 0} (\lambda I + A^T A)^{-1} A^T$ is taken to be the pseudoinverse of A as $\lambda \rightarrow 0$.

Algorithm 3 Noise-Tolerant BLS for PML

Input: Partial multi-label data matrix: $X \in \mathbb{R}^{n \times d}$, label matrix $Y \in \{0, 1\}^{k \times n}$, the number of mapped features: n , the number of enhancement nodes: m

Output: weight coefficient: θ

- 1 Derive C by algorithm 1;
 - 2 Derive X' by algorithm 2;
 - 3 **for** $i = 1 : n$ **do**
 - 4 Generating θ_{ei}, b_{ei} randomly;
 - 5 Calculate $Z_i = \varphi(X'\theta_{ei} + b_{ei})$ according to 41;
 - 6 **end**
 - 7 **for** $j = 1 : m$ **do**
 - 8 Generating θ_{hj}, b_{hj} randomly;
 - 9 Calculate $E_j = \xi([Z_1, Z_2, \dots, Z_n]\theta_{hj} + b_{hj})$ according to 42;
 - 10 **end**
 - 11 Setting $A = [Z^n | E^m]$ according to 43;
 - 12 Derive θ according to 45;
-

Algorithm 3 summarizes the framework of the proposed BLS-PML. Due to the existence of noise labels in the partial multilabel dataset, first, in step 1, the label confidence matrix is obtained by Algorithm 1, and in step 2, the feature matrix after dimension reduction is obtained by Algorithm 3. The obtained feature matrix and the label confidence matrix are jointly involved in the construction of a noise-tolerant BLS, and the final weight matrix is calculated through steps 3–12.

IV. EXPERIMENTS

To validate the feasibility of the proposed BLS-PML method, we perform extensive experiments on seven synthetic and six real-world datasets pertaining to PML. First, in Section IV-A, we outline the setup of our experiments; then, to prove the effectiveness of BLS-PML, we conduct comparative experiments with the existing PML method and perform statistical tests on the experimental results in Section IV-B1. Finally, we further analyze the model in Section IV-C, including parameter sensitivity analysis, ablation experiments, convergence analysis of the algorithm, and complexity analysis.

TABLE I

INTRODUCTION TO THE PML EXPERIMENTAL DATASETS. THE AVERAGE NUMBER OF GROUND-TRUTH LABELS (AVG. #GLS) AND THE AVERAGE NUMBER OF CANDIDATE LABELS (AVG. #CLS) ARE ALSO RECORDED FOR EACH PML DATASET

Data Set	#Examples	#Features	#Class Labels	avg.#CLS	avg.#GLs	#Domain
yeastBP	6,139	6,139	217	5.93	5.54	biology
yeastCC	6,139	6,139	50	1.39	1.35	biology
yeastMF	6,139	6,139	39	1.04	1.01	biology
mirflickr	10,433	100	7	3.35	1.77	images
music_emotion	6,833	98	11	5.29	2.42	images
music_style	6,839	98	10	6.04	1.44	images
scene	2407	294	6	2.3	1.07	images
image	2000	294	5	2.3	1.24	images
emotions	593	72	6	2.3	1.87	music
CAL_500	502	68	174	43.60,78	26.04	music
enron	351	260	19	2.46	1.86	text
yeast	2417	103	14	5.68	4.24	images
birds	351	260	19	2.46	1.86	audio

A. Experimental Setup

In this section, we begin by presenting the details of our experimental configuration, including the baseline datasets, the algorithms under comparison, and the evaluation metrics used.

1) *Datasets*: Table I provides an overview of the characteristics of the 13 real-world and synthetic PML datasets employed in the comparative studies conducted in this article. To create synthetic PML datasets, a multilabel dataset can be transformed by introducing random labeling noise. For each instance, we randomly select irrelevant labels to form the candidate label set along with the ground-truth labels. Additionally, preprocessing steps are conducted to facilitate partial labeling, excluding instances with no relevant labels. Table I showcases the details of the synthetic PML datasets generated by employing seven baseline multilabel datasets,¹ including *scene*, *image*, *emotions*, *CAL_500*, *enron*, *yeast*, *birds*. Various configurations are considered by altering the average number of candidate labels (avg. #CLS) for each multilabel dataset, resulting in a total of 18 synthetic PML datasets. Furthermore, six real-world PML datasets, namely *music_style*, *music_emotion*, *mirflickr*, *yeastBP*, *yeastCC*, and *yeastMF*² are utilized in this study [41].

2) *Learning Approaches*: To demonstrate the efficacy of the BLS-PML³ method, nine state-of-the-art methods are selected for comparison. It includes eight PML methods and one BLS-based MLL method. They are BLS_MLL [13], PML-LRS [42], CLLFS [43], PAMB [3], PARTICLE [25], PMLFS [18], PML-LCom [44], PENAD [11], and PML-NI [45]. The parameters are adopted from the default parameters in their papers.

The configuration of the parameters for our method in this study is as follows: $\lambda_1 = 1$ and $\lambda_2 = 1$. Additionally, based on experimental knowledge, the feature dimension is reduced to 50% of its original size. Moreover, fine-tuning has been performed on certain datasets, and a detailed analysis can be found in Section IV-C.

3) *Evaluation Metrics*: To assess performance, we utilize six widely used multilabel metrics in our evaluation, including average precision, ranking loss, hamming loss, one error, coverage, and macro-F1. Smaller metric values indicate better performance for the first four metrics, whereas higher values

¹<http://www.uco.es/kdis/mlresources/>

²<http://palm.seu.edu.cn/zhangml/>

³<https://markwalton-tu.github.io/>

TABLE II
PERFORMANCE OF EACH LEARNING APPROACH MEASURED BY HAMMING LOSS↓, AND BOLD THE BEST PERFORMANCE

Data Set	avg.#CLs	BLS-MLL	PML-LRS	CLLFS	PAMB	PARTICLE	PMLFS	PML-LCom	PENAD	PML-NI	Ours
mirflickr	5.93	.520±.019	.310±.018	.234±.055	.176±.031	.192±.071	.253±.003	.213±.006	.204±.058	.214±.003	.174±.055
music_emotion	1.39	.508±.131	.321±.003	.335±.005	.205±.003	.238±.004	.213±.005	.236±.014	.207±.003	.346±.033	.217±.006
music_style	1.04	.698±.140	.818±.004	.852±.005	.144±.005	.125±.004	.148±.006	.174±.021	.111±.005	.845±.007	.124±.006
YeastBP	3.35	.039±.005	.030±.002	.963±.007	.150±.056	.063±.006	.059±.002	.049±.003	.103±.005	.074±.004	.040±.004
YeastCC	5.29	.462±.107	.077±.003	.072±.006	.118±.026	.104±.011	.075±.003	.080±.006	.104±.018	.102±.003	.070±.007
YeastMF	6.04	.471±.129	.054±.001	.062±.006	.109±.038	.112±.013	.069±.004	.081±.007	.108±.006	.104±.005	.091±.007
scene	2	.527±.017	.142±.052	.207±.056	.355±.066	.165±.040	.230±.038	.194±.013	.295±.091	.167±.014	.112±.054
	3	.518±.019	.335±.028	.354±.076	.350±.094	.183±.060	.229±.037	.405±.026	.320±.066	.398±.076	.180±.061
image	2	.536±.012	.231±.007	.223±.017	.229±.026	.187±.006	.320±.050	.296±.019	.354±.010	.319±.014	.170±.019
	3	.526±.019	.724±.003	.741±.007	.206±.015	.206±.006	.359±.037	.695±.012	.196±.013	.648±.012	.238±.017
emotions	2	.498±.014	.300±.009	.236±.029	.214±.018	.220±.024	.328±.037	.261±.035	.228±.030	.327±.067	.211±.030
	3	.483±.025	.613±.014	.630±.027	.278±.022	.263±.033	.339±.025	.597±.028	.250±.028	.645±.030	.244±.040
CAL_500	43	.155±.006	.127±.001	.141±.007	.143±.006	.145±.006	.195±.010	.141±.003	.151±.007	.209±.128	.127±.004
	60	.157±.006	.137±.004	.153±.008	.144±.006	.145±.006	.223±.008	.165±.007	.154±.007	.252±.204	.137±.004
	78	.159±.006	.180±.018	.208±.010	.153±.013	.145±.006	.276±.016	.262±.008	.159±.008	.368±.234	.142±.005
enron	2	.316±.049	.108±.000	.102±.015	.125±.011	.143±.006	.162±.040	.102±.011	.144±.010	.146±.014	.091±.013
	4	.357±.111	.108±.000	.109±.017	.133±.012	.156±.013	.208±.016	.112±.025	.144±.010	.243±.018	.103±.013
	6	.392±.084	.124±.003	.117±.018	.129±.015	.167±.013	.229±.031	.130±.026	.149±.013	.342±.022	.115±.014
yeast	5	.490±.012	.202±.003	.208±.009	.255±.008	.232±.005	.303±.007	.206±.006	.253±.009	.209±.012	.187±.006
	6	.500±.006	.212±.004	.217±.010	.268±.011	.234±.005	.303±.009	.226±.006	.254±.008	.232±.011	.208±.008
	8	.501±.010	.354±.033	.395±.018	.280±.013	.234±.006	.303±.009	.440±.014	.254±.006	.462±.019	.226±.007
birds	2	.299±.081	.108±.000	.100±.012	.122±.014	.142±.008	.128±.016	.102±.009	.145±.009	.147±.013	.097±.012
	4	.409±.067	.108±.000	.094±.008	.122±.011	.156±.016	.172±.020	.113±.020	.152±.010	.243±.024	.131±.011
	6	.395±.080	.109±.001	.099±.013	.135±.016	.240±.032	.204±.033	.130±.020	.154±.011	.347±.027	.143±.012

TABLE III
PERFORMANCE OF EACH LEARNING APPROACH MEASURED BY RANKING LOSS↓, AND BOLD THE BEST PERFORMANCE

Data Set	avg.#CLs	BLS-MLL	PML-LRS	CLLFS	PAMB	PARTICLE	PMLFS	PML-LCom	PENAD	PML-NI	Ours
mirflickr	5.93	.540±.021	.166±.012	.132±.045	.129±.042	.127±.103	.454±.012	.120±.016	.130±.032	.126±.005	.101±.066
music_emotion	1.39	.466±.054	.253±.002	.288±.011	.235±.007	.360±.013	.453±.015	.463±.107	.233±.009	.238±.008	.161±.006
music_style	1.04	.553±.135	.156±.001	.213±.014	.221±.014	.218±.008	.529±.023	.399±.210	.143±.008	.140±.006	.140±.005
YeastBP	3.35	.486±.029	.199±.037	.385±.054	.592±.027	.378±.011	.363±.014	.243±.014	.172±.037	.268±.011	.120±.019
YeastCC	5.29	.496±.015	.133±.075	.223±.050	.545±.049	.317±.009	.452±.029	.207±.018	.160±.070	.225±.017	.094±.012
YeastMF	6.04	.501±.029	.137±.066	.266±.025	.478±.029	.394±.010	.416±.015	.256±.021	.189±.024	.285±.015	.062±.013
scene	2	.498±.032	.130±.075	.235±.132	.674±.161	.229±.053	.453±.019	.219±.028	.509±.182	.234±.019	.066±.033
	3	.498±.018	.144±.068	.235±.126	.445±.131	.267±.077	.478±.022	.256±.024	.494±.168	.255±.013	.141±.061
image	2	.512±.028	.186±.005	.206±.017	.281±.041	.185±.021	.452±.054	.285±.026	.517±.023	.234±.023	.096±.011
	3	.497±.021	.213±.013	.325±.026	.212±.023	.204±.024	.510±.032	.304±.033	.213±.016	.279±.028	.199±.015
emotions	2	.486±.038	.271±.006	.211±.030	.163±.033	.174±.029	.430±.064	.196±.032	.166±.026	.201±.036	.086±.015
	3	.475±.035	.304±.010	.283±.045	.256±.021	.223±.026	.479±.047	.248±.027	.213±.023	.270±.027	.160±.028
CAL_500	43	.534±.020	.177±.002	.185±.006	.280±.027	.175±.007	.440±.031	.192±.007	.226±.019	.239±.012	.096±.003
	60	.524±.011	.177±.002	.188±.007	.287±.022	.181±.007	.451±.024	.200±.011	.243±.017	.260±.011	.144±.005
	78	.520±.009	.187±.003	.198±.006	.344±.024	.194±.008	.463±.019	.213±.009	.264±.018	.284±.014	.181±.004
enron	2	.560±.058	.323±.004	.219±.049	.203±.030	.299±.022	.472±.068	.299±.056	.296±.036	.192±.031	.102±.040
	4	.516±.033	.318±.007	.301±.062	.333±.059	.311±.022	.416±.050	.303±.043	.296±.037	.242±.032	.120±.022
	6	.506±.040	.363±.006	.260±.042	.236±.031	.316±.037	.491±.047	.326±.035	.328±.036	.274±.054	.177±.034
yeast	5	.490±.012	.168±.002	.178±.014	.202±.012	.181±.011	.433±.012	.172±.008	.167±.011	.179±.010	.044±.004
	6	.498±.012	.173±.002	.177±.014	.224±.017	.184±.010	.454±.017	.177±.009	.173±.012	.187±.009	.109±.006
	8	.501±.013	.178±.001	.183±.010	.278±.036	.186±.011	.444±.019	.191±.015	.190±.010	.205±.013	.140±.006
birds	2	.546±.034	.339±.007	.215±.041	.198±.026	.281±.025	.413±.040	.310±.064	.298±.019	.187±.033	.105±.033
	4	.515±.057	.353±.009	.232±.058	.211±.039	.306±.041	.378±.031	.318±.033	.312±.044	.241±.034	.140±.029
	6	.511±.048	.321±.016	.314±.061	.297±.047	.357±.038	.390±.068	.328±.031	.327±.034	.279±.061	.193±.040

indicate better performance for the last two metrics. We conduct ten-fold cross-validation on each dataset, recording the standard deviation and mean metric value for each learning strategy. For more details, please refer to [46], [47], [48].

B. Experimental Results

The detailed experimental results for each evaluation metric and learning method can be found in Tables II–V and Tables II and III (in the Supplementary Material). These tables present the classifier’s performance for hamming loss, ranking loss, one error, coverage, average precision, and macro-F1 evaluation metrics.⁴

1) *Statistical Analysis*: Furthermore, to assess the effectiveness of various learning methods, the *Friedman test* [49] is

⁴The data for one error and macro-F1 are in the Supplementary Material.

employed. Given n learning methods and N datasets, let r_i^j represent the ranking of the j th method on the i th dataset, where the average ranking is shared in the case of a tie. Let $R_j = (1/N) \sum_{i=1}^N r_i^j$ stand for the average ranking of the j th algorithm, the Friedman test’s null hypothesis is that there are no discernible variations in how well various algorithms work, and the Friedman statistic F_F is calculated as follows, based on the F-distribution with $n - 1$ numerator degrees of freedom and $(n - 1)(N - 1)$ denominator degrees of freedom

$$F_F = \frac{(N - 1)\chi_F^2}{N(n - 1) - \chi_F^2}$$

$$\chi_F^2 = \frac{12N}{n(n + 1)} \left[\sum_{j=1}^n R_j^2 - \frac{n(n + 1)^2}{4} \right].$$

TABLE IV
PERFORMANCE OF EACH LEARNING APPROACH MEASURED BY COVERAGE↓, AND BOLD THE BEST PERFORMANCE

Data Set	avg.#CLs	BLS-MLL	PML-LRS	CLLFS	PAMB	PARTICLE	PMLFS	PML-LCom	PENAD	PML-NI	Ours
mirflickr	5.93	.725±.084	.478±.008	.376±.075	.375±.084	.381±.381	.960±.022	.371±.014	.377±.072	.374±.004	.360±.113
music_emotion	1.39	.704±.052	.501±.001	.536±.012	.500±.011	.597±.597	.965±.013	.692±.098	.493±.010	.494±.009	.393±.009
music_style	1.04	.671±.114	.311±.001	.377±.018	.381±.015	.381±.381	.935±.022	.550±.192	.304±.011	.300±.011	.292±.008
YeastBP	3.35	.711±.058	.316±.050	.676±.038	.844±.014	.644±.644	.895±.016	.450±.016	.360±.020	.488±.020	.361±.057
YeastCC	5.29	.692±.035	.239±.081	.377±.057	.710±.050	.508±.508	.922±.021	.334±.019	.287±.081	.355±.020	.236±.020
YeastMF	6.04	.641±.031	.205±.068	.382±.033	.586±.042	.519±.519	.897±.041	.346±.018	.293±.025	.387±.010	.156±.026
scene	2	.596±.023	.285±.062	.378±.109	.744±.139	.375±.042	.914±.030	.364±.026	.601±.150	.377±.016	.236±.030
	3	.597±.015	.296±.057	.377±.106	.551±.108	.407±.063	.947±.021	.396±.019	.591±.139	.395±.012	.298±.048
image	2	.655±.023	.401±.004	.422±.019	.482±.036	.405±.022	.958±.018	.480±.022	.654±.015	.442±.021	.325±.012
	3	.645±.020	.426±.011	.511±.022	.427±.019	.419±.022	.970±.011	.491±.026	.424±.015	.475±.028	.410±.015
emotions	2	.728±.048	.617±.007	.509±.048	.470±.042	.479±.032	.912±.031	.491±.028	.475±.031	.507±.036	.397±.039
	3	.722±.026	.651±.016	.589±.068	.559±.034	.523±.039	.957±.022	.546±.026	.513±.037	.560±.027	.464±.048
CAL_500	43	.979±.004	.741±.008	.776±.011	.896±.021	.730±.019	.696±.041	.813±.013	.858±.019	.871±.013	.595±.029
	60	.975±.005	.744±.005	.791±.026	.899±.017	.731±.021	.708±.021	.837±.020	.875±.015	.889±.016	.629±.025
	78	.969±.005	.789±.009	.818±.012	.937±.023	.771±.022	.695±.035	.851±.013	.885±.015	.893±.014	.570±.026
enron	2	.708±.058	.490±.007	.378±.063	.350±.043	.457±.022	.573±.052	.453±.064	.452±.041	.354±.050	.254±.075
	4	.65±.034	.488±.010	.466±.072	.486±.056	.478±.026	.554±.057	.469±.052	.457±.045	.404±.043	.248±.035
	6	.663±.044	.519±.009	.433±.046	.395±.050	.472±.056	.575±.049	.491±.047	.492±.054	.434±.072	.313±.052
yeast	5	.835±.016	.523±.002	.538±.017	.576±.017	.532±.015	.954±.020	.523±.018	.529±.017	.547±.020	.361±.008
	6	.843±.012	.531±.002	.536±.018	.603±.025	.543±.015	.952±.019	.535±.012	.542±.017	.562±.009	.440±.010
	8	.847±.011	.542±.002	.542±.019	.652±.042	.554±.015	.954±.025	.556±.023	.568±.020	.587±.018	.476±.008
birds	2	.679±.033	.510±.005	.374±.053	.342±.040	.435±.037	.552±.064	.466±.066	.453±.022	.354±.033	.249±.054
	4	.651±.053	.533±.012	.392±.070	.367±.048	.463±.042	.548±.095	.491±.052	.470±.049	.414±.055	.275±.043
	6	.657±.044	.495±.017	.492±.068	.506±.041	.514±.050	.587±.095	.494±.044	.482±.037	.440±.083	.328±.055

TABLE V
PERFORMANCE OF EACH LEARNING APPROACH MEASURED BY AVERAGE PRECISION↑, AND BOLD THE BEST PERFORMANCE

Data Set	avg.#CLs	BLS-MLL	PML-LRS	CLLFS	PAMB	PARTICLE	PMLFS	PML-LCom	PENAD	PML-NI	Ours
mirflickr	5.93	.441±.033	.773±.016	.776±.057	.778±.048	.813±.136	.476±.011	.802±.018	.777±.042	.785±.007	.823±.072
music_emotion	1.39	.387±.054	.586±.002	.559±.015	.639±.012	.507±.015	.432±.010	.428±.090	.633±.013	.624±.012	.665±.011
music_style	1.04	.305±.110	.703±.001	.660±.014	.648±.012	.657±.012	.293±.007	.497±.177	.740±.011	.737±.011	.702±.008
YeastBP	3.35	.079±.007	.287±.045	.170±.063	.063±.010	.142±.012	.195±.013	.351±.024	.395±.091	.315±.013	.696±.055
YeastCC	5.29	.147±.008	.588±.079	.496±.070	.131±.027	.278±.013	.225±.018	.512±.038	.537±.117	.477±.024	.716±.046
YeastMF	6.04	.146±.015	.553±.093	.397±.042	.201±.038	.230±.010	.248±.015	.435±.029	.429±.036	.381±.024	.724±.021
scene	2	.425±.035	.766±.119	.632±.186	.343±.090	.729±.067	.459±.024	.666±.021	.391±.191	.696±.020	.845±.081
	3	.423±.017	.752±.107	.633±.179	.405±.105	.690±.097	.441±.024	.630±.022	.376±.139	.667±.017	.716±.113
image	2	.479±.026	.763±.006	.745±.020	.699±.034	.782±.023	.521±.043	.673±.025	.477±.017	.730±.027	.836±.018
	3	.493±.015	.734±.014	.635±.026	.753±.025	.761±.024	.491±.041	.644±.042	.754±.015	.677±.020	.718±.018
emotions	2	.519±.038	.704±.006	.748±.029	.807±.030	.786±.038	.573±.050	.752±.037	.803±.036	.765±.035	.860±.019
	3	.527±.038	.689±.009	.693±.034	.725±.032	.747±.033	.528±.042	.719±.030	.761±.029	.700±.034	.786±.035
CAL_500	43	.172±.015	.506±.003	.497±.014	.419±.032	.508±.016	.219±.023	.504±.014	.467±.031	.453±.017	.631±.016
	60	.168±.010	.507±.003	.494±.014	.404±.025	.501±.015	.215±.017	.499±.018	.449±.025	.425±.016	.515±.013
	78	.163±.004	.496±.003	.487±.012	.339±.026	.495±.015	.195±.015	.485±.020	.419±.028	.389±.017	.447±.013
enron	2	.203±.029	.374±.004	.561±.069	.553±.035	.393±.052	.308±.067	.415±.055	.426±.050	.598±.044	.697±.032
	4	.227±.027	.381±.009	.444±.077	.370±.084	.390±.045	.302±.044	.414±.047	.410±.048	.513±.053	.579±.040
	6	.233±.036	.369±.010	.498±.073	.532±.039	.383±.043	.264±.042	.403±.057	.397±.043	.453±.084	.482±.049
yeast	5	.429±.010	.758±.003	.754±.015	.729±.021	.741±.014	.489±.011	.757±.016	.772±.012	.755±.016	.913±.005
	6	.424±.011	.753±.003	.753±.015	.702±.021	.738±.013	.464±.018	.755±.017	.765±.012	.749±.013	.787±.010
	8	.425±.012	.748±.002	.739±.012	.650±.033	.736±.012	.470±.020	.739±.020	.746±.012	.725±.014	.753±.012
birds	2	.197±.029	.376±.007	.567±.047	.541±.032	.417±.039	.361±.059	.405±.068	.420±.049	.610±.039	.701±.055
	4	.214±.046	.363±.006	.531±.078	.546±.056	.393±.058	.384±.035	.401±.051	.400±.053	.505±.062	.540±.061
	6	.231±.025	.383±.008	.422±.087	.315±.049	.335±.045	.353±.055	.375±.031	.363±.041	.448±.074	.451±.051

TABLE VI

SUMMARY OF FRIEDMAN STATISTICS F_F FOR EACH EVALUATION METRIC AND THE CRITICAL VALUE AT A SIGNIFICANCE LEVEL OF 0.05 IS PROVIDED FOR BLS-PML (# LEARNING METHODS $n = 10$, # DATASETS $N = 24$)

Evaluation metric	F_F	critical value
Hamming loss	15.4779	
Ranking loss	36.5027	
One error	21.0523	1.9253
Coverage	28.8607	
Average precision	26.8690	
Macro-F1	13.3023	

Table VI presents a comprehensive overview of the Friedman statistics F_F and corresponding critical values for each evaluation metric. At a significance level of 0.05, for all

evaluation metrics, the null hypothesis of similar performance among learning methods was explicitly rejected.

To illustrate the comparative performance of the learning methods, we employed the Bonferroni–Dunn test [49] as a post hoc analysis, with the BLS-PML method serving as the control. In this context, the critical difference (CD) is employed to measure the difference between the average ranks of the control approach and a specific learning method: $CD = q_\alpha(n(n+1)/(6N))^{1/2}$, where $q_\alpha = 2.773$ at 0.05 significance level and $CD = 2.4236$. As a result, if the performance of the control method and one learning method differs by at least one CD, then this is considered a significant difference.

Using BLS-PML as the control method, for each evaluation metric, the CD diagrams are illustrated in Fig. 4. The average rank of each learning strategy is displayed below the axis,

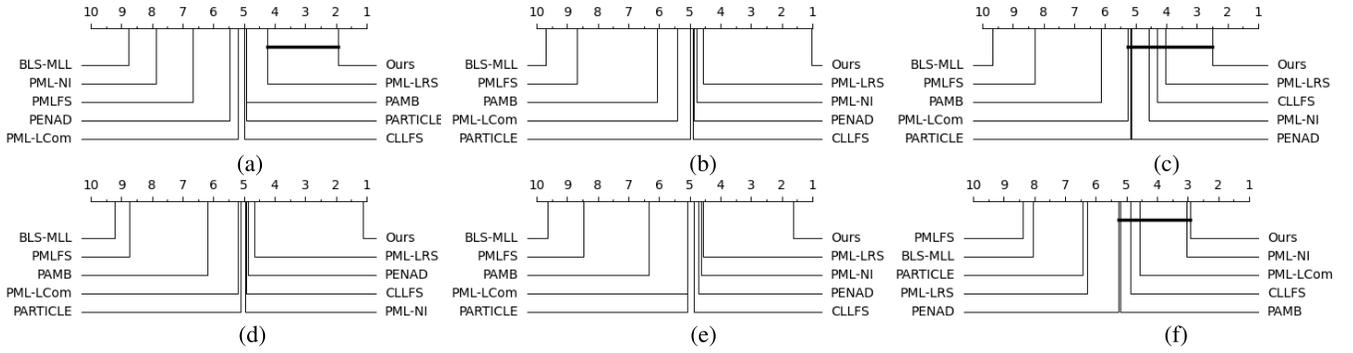


Fig. 4. Comparison results of BLS-PML (control approach) against other eight comparing approaches with the Bonferroni–Dunn test. Approaches not connected with BLS-PML by the thick horizontal line are considered to have significantly different performance with the control approach ($CD = 2.4521$ at 0.05 significance level). (a) Hamming loss. (b) Ranking loss. (c) One error. (d) Coverage. (e) Average precision. (f) Macro-F1.

TABLE VII

WIN/TIE/LOSS COUNTS OF PAIRWISE t -TEST (AT 0.05 SIGNIFICANCE LEVEL) BETWEEN BLS-PML AND EACH COMPARISON APPROACH

	BLS-PML against								
	BLS-MLL	PML-LRS	CLLFS	PAMB	PARTICLE	PMLFS	PML-LCom	PENAD	PML-NI
Hamming Loss	22/2/0	13/3/8	15/3/6	13/8/3	13/8/3	20/3/1	15/6/3	18/2/4	21/3/0
Ranking Loss	24/0/0	20/4/0	23/1/0	23/1/0	22/2/0	24/0/0	23/1/0	22/2/0	22/2/0
One Error	24/0/0	11/10/3	11/9/4	13/7/4	12/9/3	22/2/0	14/8/2	18/1/5	10/12/2
Coverage	24/0/0	19/5/0	22/2/0	23/1/0	22/2/0	24/0/0	23/1/0	21/3/0	22/2/0
Average Precision	24/0/0	16/6/2	19/4/1	21/1/2	20/2/2	24/0/0	19/4/1	21/0/3	18/5/1
Macro-F1	20/3/1	14/4/6	14/5/5	16/3/5	17/2/5	23/1/0	14/4/6	11/6/7	10/8/6
In Total	138/5/1	93/32/19	104/24/16	109/21/14	106/25/13	137/6/1	108/24/12	111/14/19	103/32/9

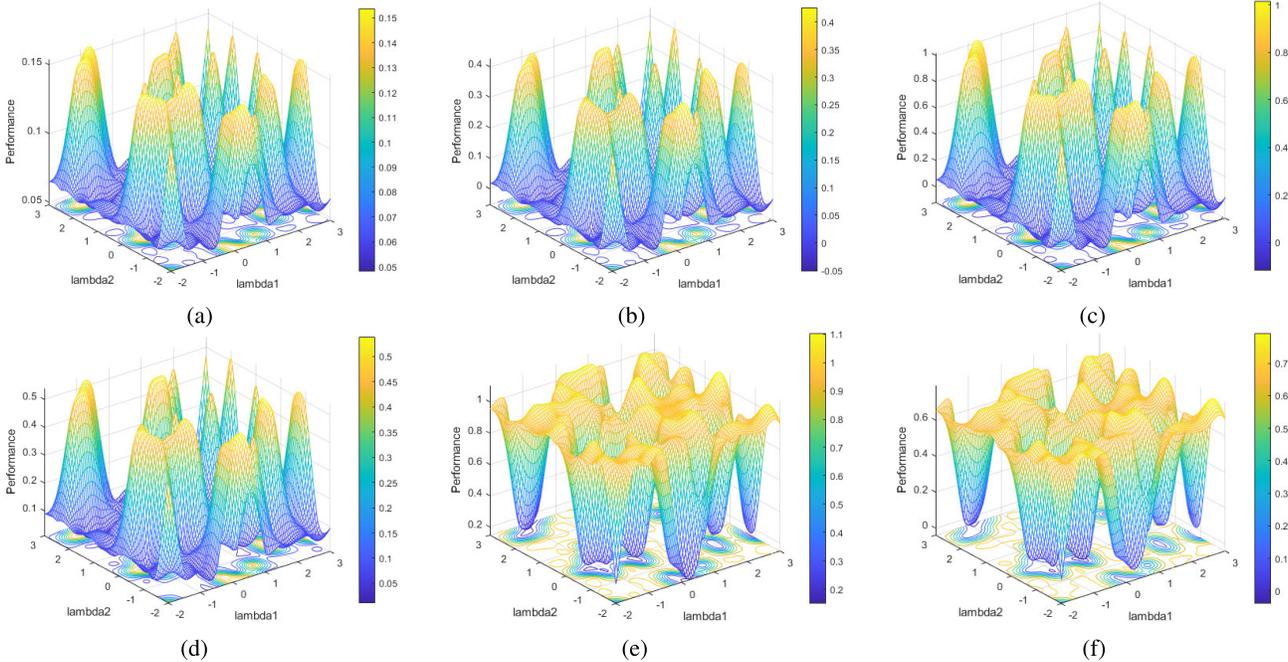


Fig. 5. Performance changes of hamming loss, ranking loss, one error, coverage, average precision, and macro-F1 as variable λ_1 and variable λ_2 are varied on yeastMF datasets. (a) Hamming loss. (b) Ranking loss. (c) One error. (d) Coverage. (e) Average precision. (f) Macro-F1.

with higher ranks toward the right. In each subfigure, a thick line connects any learning methods with average ranks within one CD of the control method, indicating similar performance. Conversely, if there is no such line, it suggests a significant difference in performance compared to the control method.

Furthermore, to assess whether the BLS-PML method exhibits significant superiority over the comparison methods

across all six evaluation metrics, a paired t -test was conducted for each dataset at a significance level of 0.05. In the paired t -test, a win (or loss) is recorded when BLS-PML significantly outperforms (or underperforms) the comparative strategy on a given dataset; otherwise, a tie is noted. There are 144 comparison pairs between the BLS-PML method and each of the eight comparison algorithms (24 PML datasets multiplied

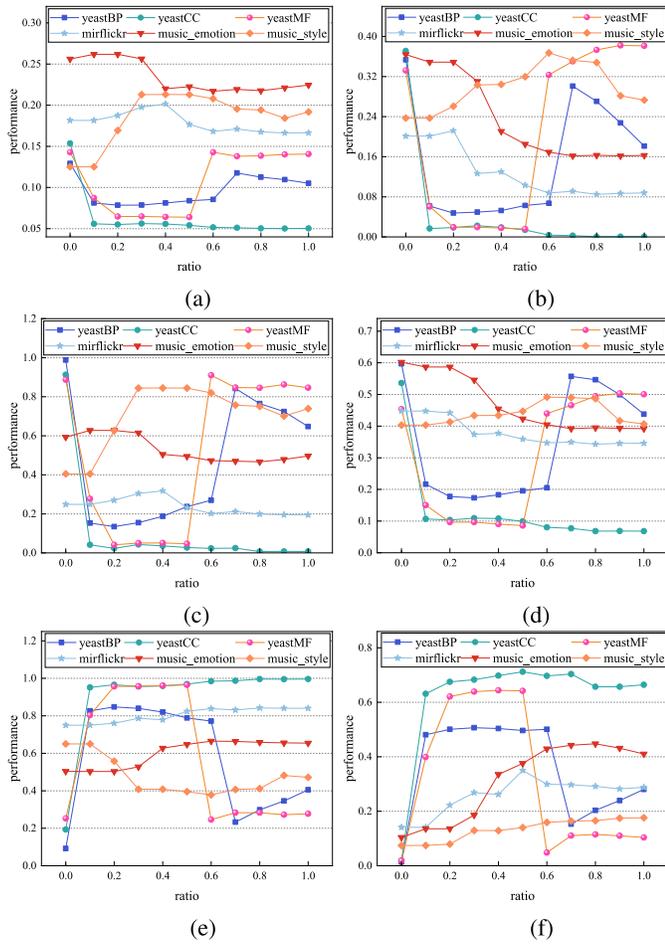


Fig. 6. Changes in hamming loss, ranking loss, one error, coverage, average precision, and macro-F1 are observed as the ratio threshold is varied across six real-world PML datasets. (a) Hamming loss. (b) Ranking loss. (c) One error. (d) Coverage. (e) Average precision. (f) Macro-F1.

by six evaluation metrics). Table VII presents the cumulative win/tie/loss for each comparison method.

According to the presented experimental results, the following observations can be deduced from the comparative study.

- 1) As shown in Tables II–V and Tables II and III (in the Supplementary Material), the proposed algorithm outperforms the comparison algorithm in over 83% of the cases for the six real datasets under the three evaluation metrics of ranking loss, coverage, and average precision, and is largely comparable to or slightly ahead of the comparison algorithm for the other three evaluation metrics. One reason for the more stable performance on synthetic datasets, such as *emotions*, *enron*, and *yeast*, is that the data distribution aligns more closely with the assumptions of the proposed method.
- 2) Fig. 4 highlights the remarkable achievement of BLS-PML, as it consistently achieves the lowest average rank across all evaluation metrics. Compared to existing PML methods, BLS-PML’s performance is statistically on par with PML-LRS in terms of hamming loss and one error, and comparable to the one error or macro-F1 of PML-NI, PML-LCom, PENAD, CLLFS, while it outperforms the comparison methods in other cases.

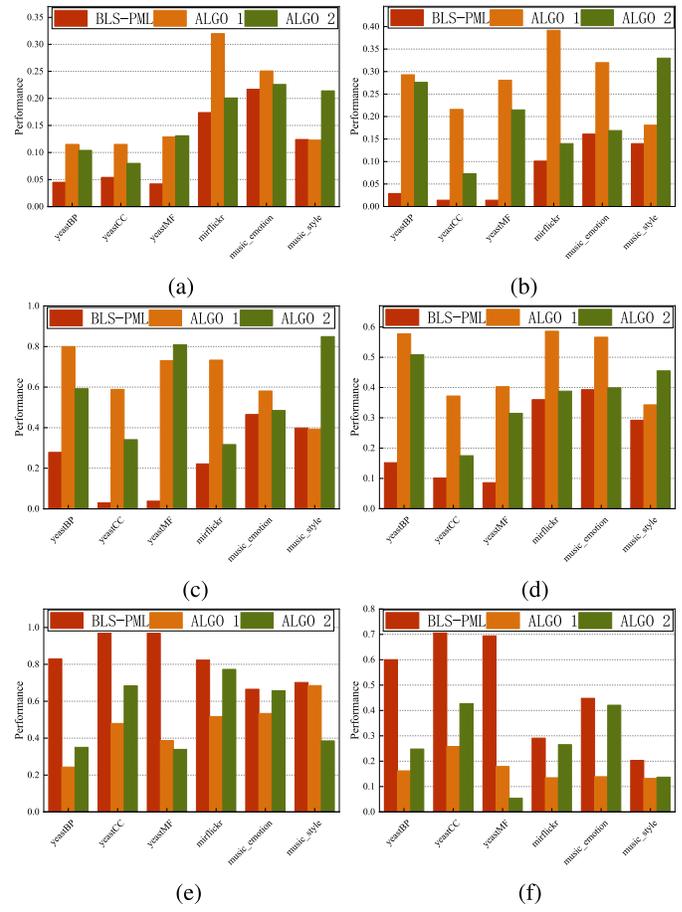


Fig. 7. Performance of BLS-PML and its simplified version (without Algorithm 1, without Algorithm 2) is evaluated on six real-world PML datasets. (a) Hamming loss. (b) Ranking loss. (c) One error. (d) Coverage. (e) Average precision. (f) Macro-F1.

- 3) As shown in Table VII, among 144 cases, the proposed method BLS-PML significantly obtains superior performance on about 96%, 65%, 72%, 76%, 74%, 95%, 75%, 77%, and 72% cases against BLS-MLL, PML-LRS, CLLFS, PAMB, PARTICLE, PMLFS, PML-LCom, PENAD, and PML-NI, respectively.

In conclusion, these findings provide strong evidence supporting the effectiveness of the proposed method for learning from PML examples.

C. Further Analysis

In this section, further experiments are performed, including parameter sensitivity analysis, ablation study, algorithm convergence analysis, and complexity analysis of the algorithm.

1) *Parameter Sensitivity*: We examine the effects of two weight parameters λ_1 and λ_2 , which balance the two regularization terms in the objective function of Algorithm 1. We conduct experiments on the yeastMF dataset and vary λ_1 and λ_2 from -2 to 3 . We evaluate the performance of the algorithm on six metrics. Fig. 5 shows that the algorithm’s performance exhibits periodic variations on the six metrics. These variations are related to the underlying sample-label associations in the dataset and indirectly validate the assumptions of sample similarity and label correlation used in Algorithm 1.

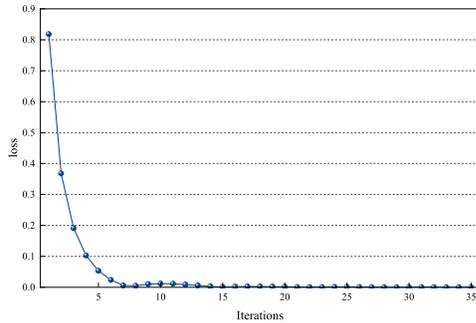


Fig. 8. Convergence trend of BLS-PML on the music_style dataset.

In Algorithm 2, the dimension of the feature space is induced to the p dimension, and p is calculated by rounding the product of the ratio and the original feature dimension. Ratio means the proportion between the value of p and the foremost feature number d . Therefore, in this chapter, the value of the ratio is changed from 0 to 1, and the changing trend of six real datasets on six evaluation indicators is observed. As can be seen from Fig. 6, for mirflickr, music emotion, and music style datasets, the basic region is stable and achieves better performance when the ratio value exceeds 0.4. For the three datasets yeastBP, yeastCC, and yeastMF, the ratio value is relatively stable between 0.1 and 0.6 and can achieve relatively superior performance. In order to reduce the redundancy of features and mitigate the impact of noise hidden in the features, the default ratio value of this experiment is 0.5.

2) *Ablation Study*: To assess the efficacy of each component within our model, we conduct an ablation study where specific methods proposed are removed, and the resulting impact on performance is measured. Specifically, in this experiment, first, the label enhancement part of Algorithm 1 is removed as a whole, and only the dimension reduction and the final BLS part are retained. Second, the label enhancement and BLS parts were retained, and the dimension reduction part was removed. The results of these three ablation experiments were compared with the results of the complete algorithm experiment.

Fig. 7, which shows the ablation results of six real PML datasets on six evaluation metrics, shows that label augmentation plays a significant role in almost all datasets, which also indicates the decomposition of noise matrix using low-rank sparsity; second, dimension reduction can also effectively improve the performance of the algorithm. When there is no label enhancement module and dimension reduction module at the same time, the performance of the algorithm has a significant decline.

3) *Convergence Analysis*: In Algorithm 1, we adopt the strategy of iterative optimization, and in order to demonstrate the convergence trend of Algorithm 1, the corresponding loss of each iteration will be counted this time.

We plot the convergence curve of Algorithm 1 on the music_style dataset in Fig. 8. The figure shows that the objective function loss decreases with the number of iterations and reaches a lower value with fewer iterations. We observe that the convergence times of the algorithm vary for different datasets, but the convergence speed is relatively fast in general.

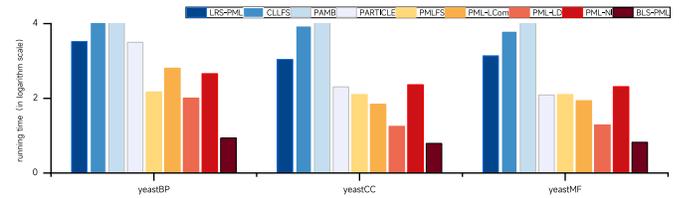


Fig. 9. Execution time (training and testing) of each learning method is measured on the three real-world PML datasets. For clarity in the histogram representation, the y-axis corresponds to the logarithm of the running time ($\log t$), with t representing the measured running time in seconds.

We set the stopping criterion as the objective function loss being less than 10^{-4} , thus ensuring the accuracy and efficiency of the algorithm on each dataset.

4) *Algorithmic Complexity*: In this article, the experiments of algorithmic time consumption are accomplished on a workstation with eight processors (2.6 GHz for each) and 64 GB RAM memory by MATLAB R2022a. Fig. 9 shows the time consumption of each algorithmic on the three real-world PML datasets. In general, the running time of the proposed algorithm is comparable to other learning methods. This benefits from the adoption of BLS, which is a fast and efficient system. At the same time, for some high-dimensional datasets, due to the dimension reduction module in the algorithm, it will be of great help to improve the efficiency of the model.

V. CONCLUSION

To balance the efficiency and accuracy, this article proposes a noise-tolerant BLS for PML, which consists of two novel layers: at the trustworthy label layer, a label enhancement method has been designed for partial multilabel data that enables label enhancement to be performed in a trustworthy label space, avoiding the bias caused by noisy labels. At the dimensionality reduction layer, to reduce the impact of ambiguity and redundancy in the feature space on the model, we designed a confidence-based kernel linear discriminant analysis method for dimensionality reduction. The experimental results show the effectiveness of the proposed method for PML. In the future, it is exciting to explore the use of advanced feature selection techniques as an alternative to dimensionality reduction to compensate for the lack of interpretability. In addition, there are partial multilabel data in fields such as medical image classification and gene prediction, and we expect that the proposed model can solve practical problems in these fields [50], [51], [52].

REFERENCES

- [1] M.-K. Xie and S.-J. Huang, "Partial multi-label learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 4302–4309.
- [2] P. Chen, J. Ye, G. Chen, J. Zhao, and P.-A. Heng, "Robustness of accuracy metric and its inspirations in learning with noisy labels," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 13, pp. 11451–11461.
- [3] B.-Q. Liu, B.-B. Jia, and M.-L. Zhang, "Towards enabling binary decomposition for partial multi-label learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13203–13217, 2023, doi: 10.1109/TPAMI.2023.3290797.
- [4] S. Xia, S. Zheng, G. Wang, X. Gao, and B. Wang, "Granular ball sampling for noisy label classification or imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2144–2155, Apr. 2023.

- [5] X. Gong, D. Yuan, and W. Bao, "Partial multi-label learning via large margin nearest neighbour embeddings," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 6, pp. 6729–6736.
- [6] F. Li, S. Shi, and H. Wang, "Partial multi-label learning via specific label disambiguation," *Knowl.-Based Syst.*, vol. 250, Aug. 2022, Art. no. 109093.
- [7] P. Zhao, S. Zhao, X. Zhao, H. Liu, and X. Ji, "Partial multi-label learning based on sparse asymmetric label correlations," *Knowl.-Based Syst.*, vol. 245, Jun. 2022, Art. no. 108601.
- [8] H. Wang, W. Liu, Y. Zhao, C. Zhang, T. Hu, and G. Chen, "Discriminative and correlative partial multi-label learning," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3691–3697.
- [9] S. He, K. Deng, L. Li, S. Shu, and L. Liu, "Discriminatively relabel for partial multi-label learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 280–288.
- [10] L. Sun, T. Wang, W. Ding, and J. Xu, "Partial multilabel learning using fuzzy neighbourhood-based ball clustering and kernel extreme learning machine," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 7, pp. 2277–2291, 2023, doi: [10.1109/TFUZZ.2022.3222941](https://doi.org/10.1109/TFUZZ.2022.3222941).
- [11] N. Xu, Y.-P. Liu, Y. Zhang, and X. Geng, "Progressive enhancement of label distributions for partial multilabel learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 4856–4867, 2023.
- [12] N. Xu, J.-Y. Li, Y.-P. Liu, and X. Geng, "Trusted-data-guided label enhancement on noisy labels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 9940–9951, 2023, doi: [10.1109/TNNLS.2022.3162316](https://doi.org/10.1109/TNNLS.2022.3162316).
- [13] J. Huang, C.-M. Vong, C. L. P. Chen, and Y. Zhou, "Accurate and efficient large-scale multi-label learning with reduced feature broad learning system using label correlation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10240–10253, 2023, doi: [10.1109/TNNLS.2022.3165299](https://doi.org/10.1109/TNNLS.2022.3165299).
- [14] J. Huang, W. Qian, C.-M. Vong, W. Ding, W. Shu, and Q. Huang, "Multi-label feature selection via label enhancement and analytic hierarchy process," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 5, pp. 1377–1393, 2023, doi: [10.1109/TETCI.2022.3231655](https://doi.org/10.1109/TETCI.2022.3231655).
- [15] W. Wei, Q. Yue, K. Feng, J. Cui, and J. Liang, "Unsupervised dimensionality reduction based on fusing multiple clustering results," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 3211–3223, Mar. 2023.
- [16] L. Sun, P. Ye, G. Lyu, S. Feng, G. Dai, and H. Zhang, "Weakly-supervised multi-label learning with noisy features and incomplete labels," *Neurocomputing*, vol. 413, pp. 61–71, Nov. 2020.
- [17] N. Xu, C. Qiao, X. Geng, and M.-L. Zhang, "Instance-dependent partial label learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 27119–27130.
- [18] J. Wang, P. Li, and K. Yu, "Partial multi-label feature selection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–9.
- [19] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.
- [20] L. Liu, J. Chen, B. Yang, Q. Feng, and C. L. P. Chen, "When broad learning system meets label noise learning: A reweighting learning framework," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–13, 2023, doi: [10.1109/TNNLS.2023.3317255](https://doi.org/10.1109/TNNLS.2023.3317255).
- [21] L. Sun, S. Feng, J. Liu, G. Lyu, and C. Lang, "Global-local label correlation for partial multi-label learning," *IEEE Trans. Multimedia*, vol. 24, pp. 581–593, 2022.
- [22] Y. Yan, S. Li, and L. Feng, "Partial multi-label learning with mutual teaching," *Knowl.-Based Syst.*, vol. 212, Jan. 2021, Art. no. 106624.
- [23] N. Cao, T. Zhang, and H. Jin, "Partial multi-label optimal margin distribution machine," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 2198–2204.
- [24] Z. Li, G. Lyu, and S. Feng, "Partial multi-label learning via multi-subspace representation," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 2612–2618.
- [25] M.-L. Zhang and J.-P. Fang, "Partial multi-label learning via credible label elicitation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3587–3599, Oct. 2021.
- [26] D. Li and Z. Zeng, "CRNet: A fast continual learning framework with random theory," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10731–10744, 2023, doi: [10.1109/TPAMI.2023.3262853](https://doi.org/10.1109/TPAMI.2023.3262853).
- [27] M. Hu, J. Heng Chion, P. N. Suganthan, and R. K. Katuwal, "Ensemble deep random vector functional link neural network for regression," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 5, pp. 2604–2615, May 2023.
- [28] Q. Shi, R. Katuwal, P. N. Suganthan, and M. Tanveer, "Random vector functional link neural network based ensemble deep learning," *Pattern Recognit.*, vol. 117, Sep. 2021, Art. no. 107978.
- [29] C. L. P. Chen, Z. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1191–1204, Apr. 2019.
- [30] G.-Y. Chen, M. Gan, C. L. P. Chen, H.-T. Zhu, and L. Chen, "Frequency principle in broad learning system," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6983–6989, Nov. 2022.
- [31] J. Jin, B. Geng, Y. Li, J. Liang, Y. Xiao, and C. L. P. Chen, "Flexible label-induced manifold broad learning system for multiclass recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2023, doi: [10.1109/TNNLS.2023.3291793](https://doi.org/10.1109/TNNLS.2023.3291793).
- [32] J. Jin, Y. Li, and C. L. P. Chen, "Pattern classification with corrupted labeling via robust broad learning system," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4959–4971, Oct. 2022.
- [33] Z. Yu, K. Lan, Z. Liu, and G. Han, "Progressive ensemble kernel-based broad learning system for noisy data classification," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9656–9669, Sep. 2022.
- [34] P. Li, B. Sheng, and C. L. P. Chen, "Face sketch synthesis using regularized broad learning system," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5346–5360, Oct. 2022.
- [35] M. Han, W. Li, S. Feng, T. Qiu, and C. L. P. Chen, "Maximum information exploitation using broad learning system for large-scale chaotic time-series prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2320–2329, Jun. 2021.
- [36] J.-W. Jin and C. L. P. Chen, "Regularized robust broad learning system for uncertain data modeling," *Neurocomputing*, vol. 322, pp. 58–69, Dec. 2018.
- [37] Y. Zheng, B. Chen, S. Wang, and W. Wang, "Broad learning system based on maximum correntropy criterion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 3083–3097, Jul. 2021.
- [38] F. Chu, T. Liang, C. L. P. Chen, X. Wang, and X. Ma, "Weighted broad learning system and its application in nonlinear industrial process modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3017–3031, Aug. 2020.
- [39] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," *Coordinated Sci. Lab., Univ. Illinois Urbana-Champaign, Tech. Rep., UILU-ENG-09-2214, DC-246*, 2009.
- [40] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Jan. 2010.
- [41] G. Yu et al., "Feature-induced partial multi-label learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 1398–1403.
- [42] L. Sun, S. Feng, T. Wang, C. Lang, and Y. Jin, "Partial multi-label learning by low-rank and sparse decomposition," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 5016–5023.
- [43] T. Xu, Y. Xu, S. Yang, B. Li, and W. Zhang, "Learning accurate label-specific features from partially multilabeled data," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2023, doi: [10.1109/TNNLS.2023.3241921](https://doi.org/10.1109/TNNLS.2023.3241921).
- [44] T. Yu, G. Yu, J. Wang, C. Domeniconi, and X. Zhang, "Partial multi-label learning using label compression," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 761–770.
- [45] M.-K. Xie and S.-J. Huang, "Partial multi-label learning with noisy label identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3676–3687, Jul. 2022.
- [46] W. Qian, J. Huang, F. Xu, W. Shu, and W. Ding, "A survey on multi-label feature selection from perspectives of label fusion," *Inf. Fusion*, vol. 100, Dec. 2023, Art. no. 101948.
- [47] W. Qian, Y. Tu, J. Qian, and W. Shu, "Partial multi-label learning via three-way decision-based tri-training," *Knowl.-Based Syst.*, vol. 276, Sep. 2023, Art. no. 110743.
- [48] W. Qian, Y. Li, Q. Ye, W. Ding, and W. Shu, "Disambiguation-based partial label feature selection via feature dependency and label consistency," *Inf. Fusion*, vol. 94, pp. 152–168, Jun. 2023.
- [49] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.
- [50] W. Gao, Y. Li, and L. Hu, "Multilabel feature selection with constrained latent structure shared term," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 3, pp. 1253–1262, Mar. 2023.
- [51] L. Ju et al., "Improving medical images classification with label noise using dual-uncertainty estimation," *IEEE Trans. Med. Imag.*, vol. 41, no. 6, pp. 1533–1546, Jun. 2022.
- [52] Y. Yang, M. Z. Hossain, E. Stone, and S. Rahman, "Spatial transcriptomics analysis of gene expression prediction using exemplar guided graph neural network," *Pattern Recognit.*, vol. 145, Jan. 2024, Art. no. 109966.



Wenbin Qian received the Ph.D. degree in computer science from the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China, in 2014.

He is currently an Associate Professor and a M.S. Supervisor with Jiangxi Agricultural University, Nanchang, China. He has authored or coauthored more than 50 papers in professional journals and conferences. His current research interests include multilabel learning and knowledge discovery.

Dr. Qian is an Associate Editor for *Journal of Intelligent and Fuzzy Systems*.



Wenhao Shu received the Ph.D. degree in computer science from the School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China, in 2015.

She is currently an Associate Professor and a M.S. Supervisor with East China Jiaotong University, Nanchang, China. Her current research interests include machine learning, granular computing, and knowledge discovery.



Yanqiang Tu received the bachelor's degree in e-commerce from Jiangxi Agricultural University, Nanchang, China, in 2021, where he is currently pursuing the master's degree in computer science and technology with the College of Computer and Information Engineering.

His current research interests include noisy learning, multilabel learning, and feature selection.



Jintao Huang received the M.S. degree from the School of Computer and Information Engineering, Jiangxi Agricultural University, Nanchang, China, in 2020, and the Ph.D. degree from the Department of Computer and Information Science, University of Macau, Macau, China, in 2023.

He is currently working as a Post-Doctoral Research Fellow at the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include machine learning, weak-supervised learning, and knowledge discovery.



Yiu-Ming Cheung (Fellow, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, in 2000.

He is currently a Chair Professor of artificial intelligence with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include machine learning and visual computing, and their applications. More details can be found at: <https://www.comp.hkbu.edu.hk/~ymc>

Dr. Cheung is a fellow of AAAS, IET, and BCS. He is the Editor-in-Chief of IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE. He also serves as an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, *Pattern Recognition*, and *Neurocomputing*.