Journal of Machine Learning Research 24 (2023) 1-48 Submitted 2/22; Revised 10/22; Published 1/23

Label Distribution Changing Learning with Sample Space Expanding

Chao Xu XCNUDT@HOTMAIL.COM
College of Science, National University of Defense Technology

Hong Tao TAOHONG.NUDT@HOTMAIL.COM
College of Science, National University of Defense Technology

Jing Zhang ZHANGJING_NUDT@163.COM
College of Science, National University of Defense Technology

Dewen Hu* DWHU@NUDT.EDU.CN
College of Intelligence Science and Technology, National University of Defense Technology

Chenping Hou* HCPNUDT@HOTMAIL.COM
College of Liberal Arts and Science, National University of Defense Technology

Editor: Russ Greiner

Abstract

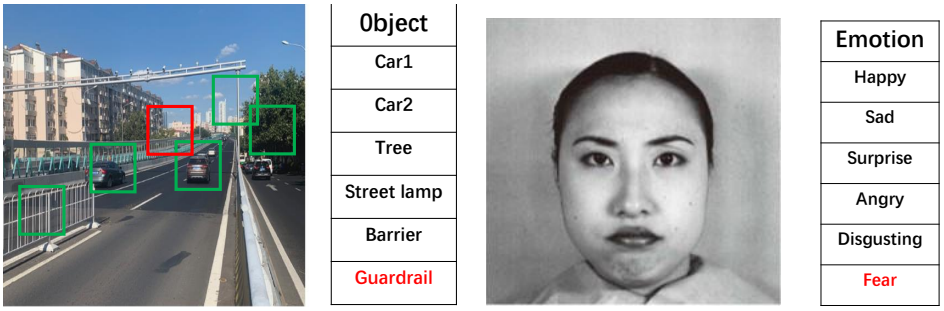
With the evolution of data collection ways, label ambiguity has arisen from various applications. How to reduce its uncertainty and leverage its effectiveness is still a challenging task. As two types of representative label ambiguities, Label Distribution Learning (LDL), which annotates each instance with a label distribution, and Emerging New Class (ENC), which focuses on model reusing with new classes, have attached extensive attentions. Nevertheless, in many applications, such as emotion distribution recognition and facial age estimation, we may face a more complicated label ambiguity scenario, i.e., label distribution changing with sample space expanding owing to the new class. To solve this crucial but rarely studied problem, we propose a new framework named as Label Distribution Changing Learning (LDCL) in this paper, together with its theoretical guarantee with generalization error bound. Our approach expands the sample space by re-scaling previous distribution and then estimates the emerging label value via scaling constraint factor. For demonstration, we present two special cases within the framework, together with their optimizations and convergence analyses. Besides evaluating LDCL on most of the existing 13 data sets, we also apply it in the application of emotion distribution recognition. Experimental results demonstrate the effectiveness of our approach in both tackling label ambiguity problem and estimating facial emotion.

Keywords: label ambiguity, label distribution learning, emerging new class

1. Introduction

Machine learning has achieved great success in many tasks, especially in supervised learning. Most of the existing approaches, such as deep learning (Fairbank et al., 2022), usually require a large amount of training data with exact logical label information. In real applications, however, we may face more learning problems with label ambiguity (Zhou, 2018; Li et al.,

*. Both Chenping Hou and Dewen Hu are the corresponding authors.



(a) (b)

Figure 1: Two typical application scenarios where new class emerges with time elapsing.

2021). For instance, due to the difficulty of labeling, incomplete, inexact and inaccurate label information may cause label ambiguity (Li et al., 2021). Among different types of label ambiguity, Label Distribution Learning (LDL) (Geng, 2016), which expresses label ambiguity by giving each instance a label distribution, and Emerging New Class (ENC) (Park and Shim, 2010), which focuses on model reusing with new classes, are two typical cases and there are some related researches (Gao et al., 2016; Mu et al., 2016).

Compared with these cases, in many real scenarios, such as emotion distribution recognition (Zhou et al., 2015) and object detection (Rudorfer, 2021), we may face a more complicated label ambiguity, i.e., label distribution changing with sample space expanding owing to the new class. For example, as shown in Fig.1, in autonomous driving, the emerging of the new label (Guardrail) will provide more comprehensive and detailed road information to improve the safety of autonomous driving. Moreover, in psychological counseling, the emerging of new micro-expressions (Fear) will provide psychologists with more information, thereby making psychological counseling more precise and efficient. Since the emerging new labels change the existing label distribution to a new label distribution, we name this new learning task as Label Distribution Changing Learning (LDCL).

To solve the LDCL problem, the most natural way is to adapt existing LDL algorithms to fit LDCL. There are three types of traditional LDL approaches. The first one transforms LDL to traditional classification task, including PT-SVM and PT-Bayes (Geng, 2016). The second type includes AA-Bayes, AA-BP (Geng, 2016) and boosting (Xing et al., 2016), which adapts the traditional algorithms to fit LDL. The final category consists of specific designed algorithms (SA) of LDL, including SA-IIS (Geng et al., 2013) and SA-BFGS (Geng, 2016). Although these approaches have achieved prominent performances to solve the LDL problem, they can not be employed to manipulate the LDCL problem directly since the existing LDL methods require consistent label information and direct adaptation of these methods will destroy the existing label distribution. In other words, if we re-normalize previous labels to a new label distribution, it will introduce label noise and degrade the performance of these methods. Besides, some new LDL algorithms are designed to solve the problem of incomplete label distribution learning (IncomLDL) (Xu and Zhou, 2017; Jia et al., 2019; Xu et al., 2021a). They are not suitable for LDCL either, since these methods do not require the existing labels to form a distribution. Instead, they make the restored labels to form a distribution. Another possible way is to extend the existing ENC algorithms to solve the

LDCL problem. Typical ENC algorithms include Multi-label learning with Emerging New Labels (MuENL) (Zhu et al., 2018b), classification under Streaming Emerging New Classes (SENC) (Mu et al., 2016) and Multi-Instance learning with Emerging Novel class (MIEN) (Wei et al., 2021). They are not suitable for LDCL either, since these methods deal with the case with single label or hard label, instead of label distribution in LDCL. Similarly, some incremental learning methods that focus on multi-class problem, such as (Cermelli et al., 2020; Rebuffi et al., 2017), will fail to handle with the special scene of LDCL due to the unique label format of LDL. In particular, some new label enhancement methods (Xu et al., 2021b; Wang et al., 2021) enhance the logical labels via leveraging the topological information of the feature space and the correlation among the labels. Another approach assists LDL by exploiting label distribution manifold (Wang and Geng, 2021). The label enhancement (Xu et al., 2021b; Wang et al., 2021) is not suitable for the learning scenario of this paper. On the one hand, our existing labels already constitute a distribution, but LE is to enhance the logical label to a distribution. On the other hand, LE can only enhance the full logical label to label distribution. In our setting, however, we only have label distribution of partial categories and aim to extend it to a new label distribution with the emering of new classes.

In this paper, to solve this interesting but rarely studied problem, we propose a new framework named Label Distribution Changing Learning (LDCL). It expands the sample space by rescaling the previous distribution and further mining the topological information of the sample space to restore the emerging new class, and then estimates the emerging label value by scaling the constraint factor. Besides, solid theoretical analysis about generalization error and convergence behavior are provided. Finally, we evaluate our methods on 13 benchmark data sets, together with a real application on emotion distribution recognition.

The main contributions of this paper have the following three points:

- To our best knowledge, it is probably the first attempt to deal with label distribution learning with emerging new label. We construct a new framework named LDCL to deal with the scenario with label sample space expanding. In addition, a smart scaling regularization is designed for the model, which not only enhances the performance of the model but also acts as a bridge to communicate the two variables $\boldsymbol{p}$ and $\mathbf{Y}$ during the optimization process. It will make the optimization process more concise.

- We give the upper bound of the generalization error for the LDCL framework, which provides solid theoretical support for the LDCL algorithm. It is worth noting that this is a new attempt in this scenario and it will be of far-reaching significance to problem understanding.

- Comprehensive experimental studies validate the effectiveness of our proposal. In addition, we also apply the LDCL algorithm in the application of emotion distribution recognition. It achieves satisfactory performance in emotion distribution recognition.

The rest of this manuscript starts with some notations in Section 2. Then the LDCL model framework is presented in detail, including the overall framework and theoretical analysis. In Section 3, we detail the analyses of two specific algorithms within the proposed LDCL framework, together with their generalization analyses and convergence analyses. In Section 4, we review the closely related work about label distribution learning, incomplete

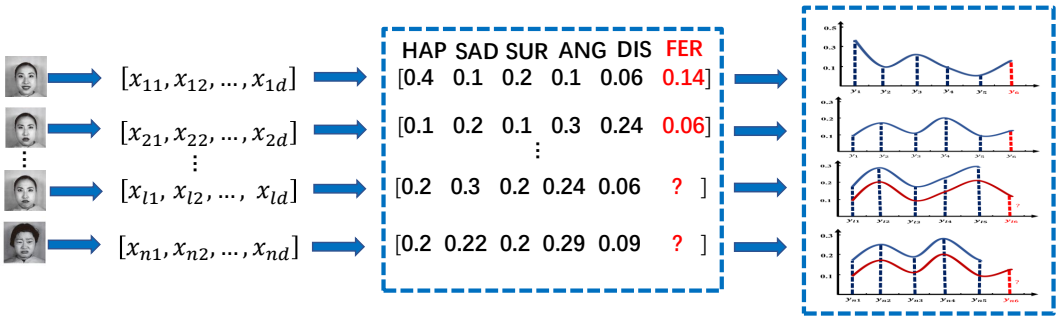


Figure 2: The specific settings of the questions raised in this article include data types and corresponding label distribution differences. In the figure, the labels marked in red are emerging new labels, and the label distribution curves marked in red are the new label distribution that need to be learned, and the label distribution curves marked in blue are the previous label distribution.

label distribution learning and emerging new class problem, together with the discussion about their differences to our method. In Section 5, we conduct experiments w.r.t. performance evaluation, parameter sensitivity and convergence behavior on data sets over various domains. Furthermore, we apply LDCL to emotion distribution recognition. Finally, we conclude this paper in Section 6.

2. The Proposed Framework

We will elaborate the settings at first and then present the corresponding framework. Finally, we deduce the generalization error bound.

2.1 Notations and Problem Setting

First of all, we give a more formal definition of LDL. Concretely, let $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ be the feature matrix of n instances $\{\boldsymbol{x}_i\}_{i=1}^n$ with dimension d. Denote its label distribution matrix as $\mathbf{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n] \in [0,1]^{c\times n}$, where c is the number of labels and $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \cdots, y_{ic}]^\top$ is the label distribution vector of $\boldsymbol{x}_i$, with y_{ij} being the description of the j-th specific label to the instance $\boldsymbol{x}_i$. In LDL, the elements of $\boldsymbol{y}_i$ is constrained to be non-negative and constitute a simplex. That is, $y_{ij} \geq 0$ and $\sum y_{ij} = 1$. It is worth noting that y_{ij} is not the probability that the j-th label correctly annotates $\boldsymbol{x}_i$, but the proportion in the complete description of $\boldsymbol{x}_i$ (Geng, 2016).

In this paper, we consider a specific LDCL setting that the label sample space is expanded, i.e., the number of interested labels is increased for the same dataset. It often occurs when the concerning target changes and increases or the annotation ability gets strong and the annotation becomes more precise. As shown in Fig.2, without loss of generality, we assume that there is a new label appearing. If there are multiple new labels, we can add a variable and add an inner loop to the optimization to handle the problem of multiple new labels appearing. Further, we assume that a few instances are manually annotated with the

new label. Actually, labeling work often has high cost and difficulty, so the amount of these manually annotated instances with the new label is often small. Thus, this LDCL setting with such assumptions is tenable.

Formally, let $\boldsymbol{y}_0 \in [0,1]^{c\times n}$ and $\hat{\mathbf{Y}} \in [0,1]^{(c+1)\times n}$ be the original and the expanded label distribution matrix, respectively. In other words, both $\boldsymbol{y}_0$ and $\hat{\mathbf{Y}}$ satisfy the constraints of label distribution matrices. For the convenience of presentation, we assume the first l instances are relabeled. Then, we have $\hat{y}_{ij} \geq 0$ and $\sum_{j=1}^{c+1} \hat{y}_{ij} = 1, i = 1, 2, \cdots, l$. For the rest $n-l$ instances, although the label space is expanding, we keep the original annotation results in the initialization, that is,

$$\hat{y}_{ij} = \begin{cases} y_{0,ij}, & 1 \leq j \leq c, \\ 0, & j = c+1, \end{cases} \quad i = l+1, \cdots, n. \tag{1}$$

Obviously, although $\hat{\mathbf{Y}}$ fulfills the constraints of the label distribution matrix, it provides no information of the new label for the last $n-l$ instances. $\hat{\mathbf{Y}}$ fails to describe the instances accurately when a new label is added into consideration. Thus, the first task of this paper is

- **Task 1:** Expanding the label sample space to accommodate the emerging new label.

Moreover, since the final aim is predicting the label distribution of the new-coming unlabeled data in real applications, our second task is

- **Task 2:** Learning a new effective classifier for the expanded label space.

In summary, given $\mathbf{X}$ and $\hat{\mathbf{Y}}$, we need to recalculate the description distribution matrix $\mathbf{Y}$ for the expanded label space and construct a classifier with $\mathbf{X}$ and $\mathbf{Y}$.

2.2 Formulation

To accomplish the above tasks, the primary challenge is predict the new label precisely while preserving the information provided by the original label distribution matrix $\mathbf{Y}_0$. To solve this problem, we denote $p_i \in (0,1)$ as the value of the new label to the instance $\boldsymbol{x}_i, i = l+1, \cdots, n$. That is, $y_{i,c+1} = p_i, i = l+1, \cdots, n$. For the label distribution vector $\boldsymbol{y}_i$ $(l+1 \leq i \leq n)$, on one hand, it satisfies

$$\sum_{j=1}^{c+1} y_{ij} = \sum_{j=1}^{c} y_{ij} + p_i = 1. \tag{2}$$

On the other hand, $y_{ij} (1 \leq j \leq c)$ should keep the original label distribution information as much as possible. A natural way is to keep the ratio between any pair of original labels unchanged, which is equivalent to compress the label values of the original label with a same ratio r_i, i.e., $y_{ij} = r_i y_{0,ij} = r_i \hat{y}_{ij} (1 \leq j \leq c)$. Thus, we have

$$\sum_{j=1}^{c} r_i \hat{y}_{ij} + p_i = r_i \sum_{j=1}^{c} \hat{y}_{ij} + p_i = 1. \tag{3}$$

Recall that $\sum_{j=1}^{c} \hat{y}_{ij} = \sum_{j=1}^{c} y_{0,ij} = 1$, then we have

$$r_i = 1 - p_i, i = l+1, \cdots, n. \tag{4}$$

That is to say, when the ratio between original labels are kept, the value of the new label equals to 1 minus the compression ratio of the original labels. For compact representation, introduce the indicator vector $\boldsymbol{\Omega} \in \{0,1\}^{(c+1)\times 1}$, where

$$\Omega_j = \begin{cases} 1, & 1 \le j \le c, \\ 0, & j = c+1. \end{cases} \tag{5}$$

Then, together with the constraints that $\boldsymbol{y}_i^\top \mathbf{1}_{c+1} = 1$, it holds that

$$(1-p_i)\boldsymbol{\Omega} \odot \hat{\boldsymbol{y}}_i = \boldsymbol{\Omega} \odot \boldsymbol{y}_i, 0 < p_i < 1, i = l+1, \cdots, n, \tag{6}$$

where $\odot$ is the Hadamard (element-wise) product (Craigen et al., 1992). $\mathbf{1}_{c+1}$ stands for the $(c+1)$-dimensional vector with all ones. In the remainder of this paper, the subscript of such kind of vectors or matrices is omitted when there is no ambiguity.

Based on the above analysis, it is known that the the expanded label distribution matrix is obtained once $\boldsymbol{p} = [p_{l+1}, p_{l+2}, \cdots, p_n]^\top$ is determined. Therefore, to obtain a precise expanded label distribution matrix and henceforth learn a qualified classifier, it is vital to integrate the instance structural information, label correlations and the prior information of the new label appropriately. With the above considerations, we propose the Label Distribution Changing Learning (LDCL) framework as follows.

$$\begin{array}{ll} \min\limits_{\mathbf{Y},\mathbf{M},\boldsymbol{p}} & \mathcal{L}(\mathbf{Y}) + \lambda\mathcal{H}(\mathbf{M},\mathbf{Y}) + \gamma\mathcal{R}(\boldsymbol{p}) \\ \text{s.t.} & (1-p_i)\boldsymbol{\Omega} \odot \hat{\boldsymbol{y}}_i = \boldsymbol{\Omega} \odot \boldsymbol{y}_i, 0 < p_i < 1, i = l+1, \cdots, n \\ & \mathbf{Y}\mathbf{1}_{c+1} = \mathbf{1}_n, \mathbf{Y} \ge \mathbf{0}, \end{array} \tag{7}$$

where $\mathbf{Y} \in [0,1]^{(c+1)\times n}$ is the expanded label distribution matrix we need to recalculated. $\mathbf{M}$ is coefficients of the learned classifier from $\mathbf{X}$ to $\mathbf{Y}$. $\mathbf{0}$ is a zero matrix with the same size of $\mathbf{Y}$, and $\mathbf{Y} \ge \mathbf{0}$ indicates that all the elements of $\mathbf{Y}$ are non-negative.

In this framework, $\mathcal{L}(\cdot)$ and $\mathcal{H}(\cdot)$ are empirical risks for calculating the expanded label distribution matrix and learning a classifier, respectively. $\mathcal{R}(\cdot)$ is a regularization term, which is encoded with the prior information of the emerging new label. Balanced by the trade-off parameters $\lambda > 0$ and $\gamma > 0$, three terms work together to realize Task 1 and Task 2 comprehensively. Specifically, the transformed feature information encoded in $\mathcal{H}(\cdot)$ helps with the label expanding process. In turn, meticulously learned label distribution matrix by optimizing $\mathcal{L}(\cdot)$ and $\mathcal{R}(\cdot)$ directs the construction of the classifier. Different choices of $\mathcal{L}(\cdot)$, $\mathcal{H}(\cdot)$ and $\mathcal{R}(\cdot)$ lead to different algorithms. In the following, the possible options of the three terms will be introduced.

As for the first term $\mathcal{L}(\mathbf{Y})$, it is the empirical risk for computing the expanded label distribution matrix $\mathbf{Y}$. To achieve more accurate computation, we consider instance structural information and label correlations simultaneously. On the whole, we assume that label correlations can be reflected by the topological relationship of instances. Specifically, as shown in Fig.3, a similarity graph is constructed to represent the topological relationship of

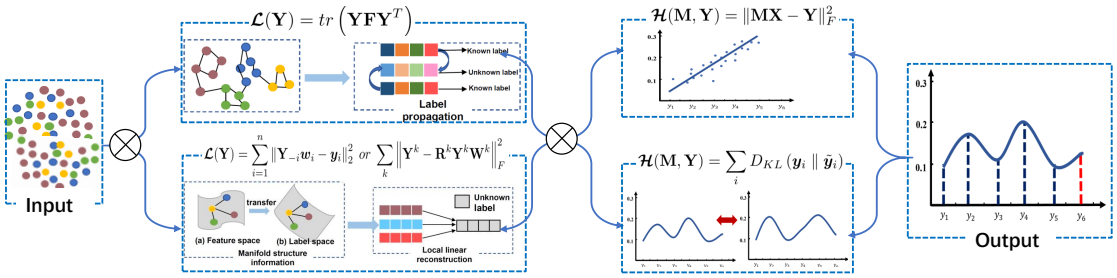


Figure 3: The basic idea of the method proposed in this article, including three sample-information-based methods. The top corresponds to the label propagation method based on similar graph, the bottom corresponds to manifold-based self-expression, i.e. manifold methods, and manifold based and embedded label correlation self-expression, i.e. manifold enhancement.

instances, based on which the relationship between labels is depicted with certain assumptions. In this way, the instance structural information is transferred from the feature space to the expanded label space.

In technical detail, we can borrow relevant techniques from semi-supervised learning paradigms such as label propagation (LP) (Li et al., 2015) and manifold learning (Hou et al., 2016). As shown in the upper left part of Fig.3, with the local invariance assumption, LP constructs a label propagation matrix based on the correlation between instances. It uses the difference in path weights in the propagation process to naturally produce differences in the description of different labels, thereby reflecting the relationship between labels contained in the training data. Hence, $\mathcal{L}(\mathbf{Y})$ can be formulated as $\mathcal{L}(\mathbf{Y}) = \mathrm{Tr}(\mathbf{Y}\mathbf{F}\mathbf{Y}^\top)$, where $\mathbf{F} \in \mathbb{R}^{n\times n}$ is the graph Laplacian matrix of the feature space. Specifically, the calculation of $\mathbf{F}$ will be given in detail in Algorithm 1.

From another aspect, based on the self-expression assumption, as displayed in the lower left part of Fig.3, the feature manifold is represented by graphs and approximated by overlapped blocks of local linear neighborhoods. Then the topological structure of the feature space is transferred to the label space, that is, the same local linear reconstruction coefficient matrix is shared. To this end, we construct a weighted graph $G(\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}}, \mathbf{W})$ (Lv et al., 2019), where $\boldsymbol{\mathcal{V}}$ is the vertex set corresponding to the training instances, $\boldsymbol{\mathcal{E}}$ is the sparsely connected edge set, and $\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_n] \in \mathbb{R}^{n\times n}$ is the weight matrix encoding the structural information to characterize the underlying structure of the feature space. Note that the diagonal element of $\mathbf{W}$ are zeros, and W_{ij} is regarded as the influence of $\boldsymbol{x}_j$ over $\boldsymbol{x}_i$. In the feature space, the reconstruction coefficient matrix $\mathbf{W}$ can be obtained column-wisely by optimizing

$$\min_{\mathbf{W}} \|\mathbf{X}_{-i}\boldsymbol{w}_i - \boldsymbol{x}_i\|_2^2 + \alpha \|\boldsymbol{w}_i\|_1 .$$

where $\boldsymbol{w}_i$ is the i-th column of $\mathbf{W}$, $\mathbf{X}_{-i}$ represents the feature matrix with its i-th instance being replaced by a zero vector and $\alpha > 0$ is a parameter. If we represent the labels of all instances one by one, then $\mathcal{L}(\mathbf{Y}) = \sum_{i=1}^{n} \|\mathbf{Y}_{-i}\boldsymbol{w}_i - \boldsymbol{y}_i\|_2^2$, where $\mathbf{Y}_{-i}$ is defined in the same manner with $\mathbf{X}_{-i}$.

Furthermore, the label correlations conduce to the recovery of the expanded label distribution matrix $\mathbf{Y}$. Intuitively, the correlative labels tend to have similar description degrees. Inspired by the investigation of (Zhu et al., 2018a), the label correlation matrix may vary among regions. Then, to learn the expanded label distribution matrix $\mathbf{Y}$ more accurately, we learn the expanded label distribution $\mathbf{Y}$ block by block, and each block corresponds to a cluster. First, the training set is divided into K regions $\left\{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^K\right\}$ by k-means clustering. $\mathbf{X}^k$ collects the instances belonging to the k-th clustering, and $\mathbf{Y}^k$ is the corresponding label distribution sub-matrix. Correspondingly, $\mathbf{W}^k$ characterizes the manifold structure information of the k-th regions, and introduce $\mathbf{R}^k$ to encode label correlation information. Borrowing the idea of label enhancement (Lv et al., 2019), we think that integrating feature structure and label correlation is effective in recovering the label distribution matrix. Therefore, we synthesize the feature structural information and the label correlations to reconstruct the expanded label distribution matrix $\mathbf{Y}$, $\mathbf{Y}^k$ is characterized by $\mathbf{Y}^k \approx \mathbf{R}^k\mathbf{Y}^k\mathbf{W}^k$, which leads to the following function to minimize:

$$\mathcal{L}(\mathbf{Y}) = \sum_k \left\| \mathbf{Y}^k - \mathbf{R}^k\mathbf{Y}^k\mathbf{W}^k \right\|_F^2,$$

Here, $\mathbf{R}^k \in \mathbb{R}^{(c+1)\times(c+1)}$, the correlation matrix, plays the role of encoding the label correlations. It has large value if the two class labels has high correlation. When we minimize the loss, it will enforce the high correlation between the two correponding columns of $\mathbf{Y}^k$, which conduces to the recovery of the expanded label distribution matrix $\mathbf{Y}$. However, constructing sub-Laplacian $\mathbf{R}^k$ from training set with new label directly is noisy. Therefore, instead of specifying any label correlation matrix, we optimize the Laplacian matrices together with the expanded label distribution matrix $\mathbf{Y}^k$ iteratively. Due to the introduction of Laplacian sub-matrix, this strategy further considers the correlation between label classes, which we call it as manifold enhancement in our learning scenario.

In short, three kinds of reconstruction losses are suggested. Thus, the first part can be formulated as but not limited to the following forms

$$\mathcal{L}(\mathbf{Y}) = \begin{cases} \mathrm{Tr}(\mathbf{Y}\mathbf{F}\mathbf{Y}^\top) \\ \sum_{i=1}^n \left\| \mathbf{Y}_{-i}\boldsymbol{w}_i - \boldsymbol{y}_i \right\|_2^2 \\ \sum_k \left\| \mathbf{Y}^k - \mathbf{R}^k\mathbf{Y}^k\mathbf{W}^k \right\|_F^2 \end{cases}. \tag{8}$$

As for the second term $\mathcal{H}(\mathbf{M}, \mathbf{Y})$, it takes charge to train a classifier based on the expanded label distribution matrix. A simple yet efficient choice for this term is to fit the expanded label distribution matrix linearly via least squares regression, that is, $\mathcal{H}(\mathbf{M}, \mathbf{Y}) = \|\mathbf{M}\mathbf{X} - \mathbf{Y}\|_F^2$. For the predicted $\mathbf{Y'} = \mathbf{M}\mathbf{X}_{\boldsymbol{t}}$, each label vector is normalized to be a distribution. Here, two normalization methods softmax transformation, i.e, $y_{ij} = e^{y'_{ij}} / \sum_j e^{y'_{ij}}$ and linear normalization (LN), i.e, $y_{ij} = y'_{ij} / \sum_j y'_{ij}$ are mainly used in this paper. It should be noted that when LN is adopted, the negative prediction is replaced by zero. In addition, KL divergence is a commonly used method to measure the difference between distributions. Specifically, the KL divergence between the predicted label distribution vector $\tilde{\boldsymbol{y}}_i$ and the recalculated expanded label distribution vector $\boldsymbol{y}_i$ is $D_{KL}\left(\boldsymbol{y}_i \parallel \tilde{\boldsymbol{y}}_i\right) = \boldsymbol{y}_i^\top \ln(\boldsymbol{y}_i / \tilde{\boldsymbol{y}}_i)$. Here $\tilde{\boldsymbol{y}}_i$

Table 1: Components of the LDCL framework.

$\mathcal{L}(\mathbf{Y})$	$\mathcal{H}(\mathbf{M},\mathbf{Y})$	$\mathcal{R}(\boldsymbol{p})$
$\mathrm{Tr}\left(\mathbf{Y}\mathbf{F}\mathbf{Y}^\top\right)$	$\|\mathbf{M}\mathbf{X}-\mathbf{Y}\|_F^2$	$\|\boldsymbol{p}\|_1$
$\sum_{i=1}^n \|\mathbf{Y}_{-i}\boldsymbol{w}_i-\boldsymbol{y}_i\|_2^2$	$\sum_i D_{KL}\left(\boldsymbol{y}_i\|\tilde{\boldsymbol{y}}_i\right)$	$\|\boldsymbol{p}\|_2^2$
$\sum_k \left\|\mathbf{Y}^k-\mathbf{R}^k\mathbf{Y}^k\mathbf{W}^k\right\|_F^2$		$-\boldsymbol{p}^\top\log\boldsymbol{p}$

is obtained by the maximum entropy model

$$\tilde{y}_{ij}=p(y_{ij}|\boldsymbol{x}_i;\mathbf{M})=\frac{1}{S_i}\exp\left(\sum_r M_{jr}x_{ir}\right),$$

where $\mathbf{M}$ is the regression coefficients and $S_i=\sum_j\exp(\sum_r M_{jr}x_{ir})$ is a normalization factor. Henceforth, the possible options for the second term can be

$$\mathcal{H}(\mathbf{M},\mathbf{Y})=\begin{cases}\|\mathbf{M}\mathbf{X}-\mathbf{Y}\|_F^2\\ \sum_i D_{KL}\left(\boldsymbol{y}_i\parallel\tilde{\boldsymbol{y}}_i\right)\end{cases}. \tag{9}$$

In order to further analyze the two classifiers, we conducted a comparative experiment experiments on these two classification models. The experiment results are shown in Table 10 Appendix D.1. From the experiment results, it can be seen that the maximum entropy model ("KL+softmax") has the best performance. Compared with "KL + softmax", the performance of "KL + LN" is reduced. Similarly, the performance of "L_2+softmax" is degenerated compared with "L_2 + LN". Therefore, compared with the wide applicability of linear regression model, the maximum entropy model with KL divergence is more targeted for LDL. On the other hand, within this framework, specific methods can be tailored and solved according to practical demand. Here we use the L_2 loss for the purpose of solving convenience. When the application scenario has stricter requirements on the prediction accuracy, KL divergence can be used. Comprehensively comparing the pros and cons of the two methods, we take both hypothetical models as candidates for $\mathcal{H}(\mathbf{M},\mathbf{Y})$.

The regularization term $\mathcal{R}(\boldsymbol{p})$, which specifies the prior information of the new label, could have different forms according to various types of prior label information. For example, if the distribution of the new label over instances obeys a normal distribution, then the L_2-norm regularization $\|\boldsymbol{p}\|_2^2$ is employed. For the case that the new label is sparsely distributed, $\mathcal{R}(\boldsymbol{p})$ can be formulated as $\|\boldsymbol{p}\|_1$. Moreover, with the consideration that the amount of information brought by the new label is limited, then the information entropy (Amjad, 2019) of the new label should be constrained to be within a certain range. That is, $\mathcal{R}(\boldsymbol{p})=-\boldsymbol{p}^\top\log\boldsymbol{p}$. Since the most common prior information can be characterized by the above three cases or their combinations, we narrow our focus on their corresponding regularizations. That is,

$$\mathcal{R}(\boldsymbol{p})=\begin{cases}\|\boldsymbol{p}\|_1\\ \|\boldsymbol{p}\|_2^2\\ -\boldsymbol{p}^\top\log\boldsymbol{p}\end{cases}. \tag{10}$$

Table 1 summarizes the suggested forms for each term and provides 18 possible LDCL models. The main idea and procedure of the proposed LDCL framework is depicted in

Fig.3. It can be seen that different combinations of $\mathcal{L}(\cdot)$, $\mathcal{H}(\cdot)$ and $\mathcal{R}(\cdot)$ lead to distinct LDCL models. In summary, in terms of model design, under the guidance of the LDCL framework presented in Eq.(7), more models can be designed to meet different purposes.

One point should be highlighted. The formula in Eq.(7) called a framework not only due to its capability for LDCL model design but also owing to its theoretical results for algorithm generalization shown in the next subsection.

2.3 Generalization ability of LDCL framework

In order to provide theoretical support for the LDCL framework, we deduce the theoretical generalization error bound analysis for LDCL framework, including the generalization error bounds of transductive and inductive mapping hypothesis families, which are both integrated in the model. The inductive model h maps $\mathbf{X}$ to $\mathbf{Y}$ with transformation matrix $\mathbf{M}$, that is, $h : \mathrm{X} \xrightarrow{\mathbf{M}} \mathrm{Y}$. The goal of the inductive model h is to predict the label of unseen test data, and the corresponding theory is Theorem 5, which depicts the prediction ability of the inductive model h. The transductive model f uses the existing feature $\mathbf{X}$ and the incomplete label distribution matrix $\hat{\mathbf{Y}}$ to reconstruct the expanded label distribution matrix $\mathbf{Y}$, that is, $f : \mathrm{X} \times \hat{\mathbf{Y}} \to \mathbf{Y}$. The goal of the transductive model f is to reconstruct the expanded label distribution of un-relabeled data, and the corresponding theory is Theorem 4, which depicts the reconstruction ability of the transductive model f.

For the transduction model that cannot establish a clear predictive model, new data cannot be directly accepted, but it performs well in restoring unrelabeled data labels. For the inductive model, an explicit model can be trained to directly predict the unseen test data, but when predicting un-relabeled data, the existing labels of un-relabeled data are ignored, so the recovery performance of the un-relabeled data labels may be poor. Therefore, considering the advantages of both, we combine both mappings into the proposed model to handle different tasks. It needs to be pointed out that the mapping space corresponding to different models under the LDCL framework is different, so the Rademacher Complexity is different, and the tightness of the generalization is also different. As a result, the model combination under the framework has different characteristics, and will show different performance for different data sets. First of all, since there is little research about the generalization error bound in LDL, we introduce the analogs as in traditional supervised learning for our analyses.

Definition 1 (Empirical and generalization risk) *In the following, we define the empirical error and generalization error based on a loss function $\ell : \mathbb{R}^{(c+1)} \times \mathbb{R}^{(c+1)} \to \mathbb{R}_+$, which measures the difference between two distributions.*

$$err_{(\boldsymbol{x},\boldsymbol{y})\sim\boldsymbol{\mathcal{D}}}(h) = E_{(\boldsymbol{x},\boldsymbol{y})\sim\boldsymbol{\mathcal{D}}}\ell(h(\boldsymbol{x}),\boldsymbol{y}),\ \hat{err}_{l+u}(h) = \frac{1}{l+u}\sum_{i=1}^{l+u}\ell(h(\boldsymbol{x}_i),\boldsymbol{y}_i),$$

$$\hat{err}_{l+u}(f) = \frac{1}{l+u}\sum_{i=1}^{l+u}\ell(f(\boldsymbol{x}_i),\boldsymbol{y}_i),\ \hat{err}_{u}(f) = \frac{1}{u}\sum_{i=l+1}^{l+u}\ell(f(\boldsymbol{x}_i),\boldsymbol{y}_i),$$

$$\hat{err}_{l}(f) = \frac{1}{l}\sum_{i=1}^{l}\ell(f(\boldsymbol{x}_i),\boldsymbol{y}_i).$$

Here, $h \in \mathcal{H}$ is the inductive mapping and $err_{(\boldsymbol{x},\boldsymbol{y})\sim\boldsymbol{\mathcal{D}}}(h)$ is the generalized error of the classifier h. $\hat{e}rr_{l+u}(h)$ is the empirical error. As for the transductive mapping $f \in \mathcal{F}$, $err_{l+u}(f)$ is full sample empirical error of the hypothesis f, $err_u(f)$ is the transduction empirical error of unlabeled data of f. And $err_l(f)$ is the transduction empirical error of labeled data of f.

Definition 2 (Rademacher Complexity (Bartlett and Mendelson, 2002)) *Given a function class $\mathcal{H}$ and a loss function $\ell : \mathbb{R}^{(c+1)} \times \mathbb{R}^{(c+1)} \to \mathbb{R}_+$. For a function $h \in \mathcal{H}$ and a sample $\mathbf{Z} = (Z_1, Z_2, \cdots, Z_n)$ of size n, $\mathbf{Z} \in \boldsymbol{\mathcal{Z}} = (\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}})$. Then, the empirical Rademacher complexity of $\mathcal{H}$ with respect to the sample $\mathbf{Z}$ is defined as*

$$\hat{\Re}_n(\ell \circ \mathcal{H} \circ \mathbf{Z}) = E_\sigma \left[\sup_{h\in\mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell(h(\boldsymbol{x}_i), \boldsymbol{y}_i) \right]. \tag{11}$$

where the random variables $\{\sigma_i\}_{i=1}^n \sigma_i$ are called Rademacher variables, which obey the uniform distribution on $\{-1, +1\}$. The Rademacher complexity of $\mathcal{H}$ is the expectation of Rademacher complexity based on the experience of all samples of size n drawn by $\mathcal{D}$

$$\Re_n(\ell \circ \mathcal{H} \circ \mathbf{Z}) = E_{\mathbf{Z}\sim\mathcal{D}^n} \left[\hat{\Re}_n(\ell \circ \mathcal{H} \circ \mathbf{Z}) \right]. \tag{12}$$

Definition 3 (Transductive Rademacher Complexity (El-Yaniv and Pechyony, 2009)) *Given a function class $\mathcal{F}$ and a loss function $\ell : \mathbb{R}^{(c+1)} \times \mathbb{R}^{(c+1)} \to \mathbb{R}_+$, for a function $f \in \mathcal{F}$ and a sample $\mathbf{Z} = (Z_1, Z_2, \cdots, Z_n)$ of size n, $Z_i = (\boldsymbol{x}_i, \boldsymbol{y}_i)$, $\mathbf{Z} \in \boldsymbol{\mathcal{Z}} = (\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}})$. Random partitioning of n points into two disjoint sets of m_1 and m_2 points. The following quantity is called transductive Rademacher complexity (TRC)*

$$\Re_{l+u}^{Td}(\ell \circ \mathcal{F} \circ \mathbf{Z}) = \left(\frac{1}{m_1} + \frac{1}{m_2} \right) E_\sigma \left[\sup_{f\in\mathcal{F}} \sum_{i=1}^{n} \sigma_i \ell(f(\boldsymbol{x}_i), \boldsymbol{y}_i) \right], \tag{13}$$

where $\sigma = \{\sigma_i\}_{i=1}^{m_1+m_2}$ are i.i.d. random variables taking values ± 1 with probabilities of P and 0 with a probability of $1 - 2P$, and $P \in \left[0, \frac{1}{2}\right]$.

After introducing the basic definition, we give the generalization error bound for the LDCL framework as follows:

Theorem 4 *Let $\mathcal{F}$ be the family of transductive mapping functions for LDCL framwork and a loss function ℓ with Lipschitz constant L_ℓ and bounded by a constant B. Given $\mathbf{Z} = \mathbf{Z}_l \cup \mathbf{Z}_u$ as a the full sample, where $\mathbf{Z}_l$ represents the sample set annotated by the new label distribution, and $\mathbf{Z}_u$ represents the sample set annotated by the previous label distribution. Define $Q \triangleq \frac{l+u}{(l+u-1/2)(1-1/(2\max(l,u)))}$ and $P \triangleq \frac{lu}{(l+u)^2}$. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ the following bound holds for all $f \in \mathcal{F}$,*

$$\hat{e}rr_u(f) \le \sqrt{2} L_\ell (c+1) \Re_{l+u}^{Td}(\mathcal{F} \circ \mathbf{Z}) + B \left(\frac{1}{l} + \frac{1}{u} \right) \sqrt{\frac{32 \ln(4e)}{3}} \sqrt{\min(l, u)} + B \sqrt{\frac{Q}{2} \left(\frac{l+u}{lu} \right) \ln\left(\frac{1}{\delta} \right)}. \tag{14}$$

Theorem 5 *Let $\mathcal{H}$ be the family of inductive mapping functions for LDCL framwork and a loss function ℓ with Lipschitz constant L_ℓ and bounded by a constant B. Given $\mathbf{Z} = \mathbf{Z}_l \cup \mathbf{Z}_u$ as a the full sample, such that $n = l + u$. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ the following bound holds for all $h \in \mathcal{H}$,*

$$err_{(\boldsymbol{x},\boldsymbol{y})\sim\boldsymbol{\mathcal{D}}}(h) \leq \hat{err}_{l+u}(h) + \hat{err}_u(f) + 2L_\ell(c+1)\Re_n(\mathcal{H}\circ\mathbf{Z}) + \frac{2L_\ell(c+1)u}{l+u}\Re_u(\mathcal{F}\circ\mathbf{Z}) + B\sqrt{\frac{\log(\frac{1}{\sigma})}{2(l+u)}}. \tag{15}$$

Due to the limitation of space, the details are presented in the Appendices. In the proof, the main differences between our procedures and the traditional steps are data type and model composition. Regarding the data type, in this scenario, the data is composed of relabeled and un-relabeled data, which is different from the traditional form with all labeled data. Thus, we employ transductive Rademacher complexity to characterize the function space complexity of the mapping to obtain the generalization error bound. Regarding the model composition, our model combines transductive and inductive mappings. Since the generalization error bound of the inductive mapping cannot be derived from the transductive setting, the Cauchy inequality is used to connect two mappings and the generalization error bound of the inductive mapping is further given.

In addition, as seen from Theorems 4 and 5, we can get some observations about the proposed model. On one hand, Theorem 4 indicates that, with the increase of labeled samples, the transductive risk upper bound of the transductive mapping decreases, which is consistent with intuition. On the other hand, according to the Cauchy inequality, the data used to train the inductive mapping contains errors from the transductive mapping. Therefore, the generalization bound of the inductive mapping may be looser than the transductive risk upper bound of the transductive mapping. Our experimental results also verify this phenomenon, the performance of the transductive model f under setting 1 for the un-relabeled data is better than that of the inductive model h under setting 2 for unseen test data, since the un-relabeled data has incorporated into the training process.

3. Model Analysis and Optimization

In order to verify the effectiveness of the LDCL framework, we choose two specific models, taking into account the differences and representativeness of the selected methods. Then we analyze these two methods in detail, including the components of the model and optimization algorithms.

3.1 Algorithm 1: Graph

For Algorithm 1, we use LP technique, and choose KL-divergence as the empirical risk for learning a classifier and information entropy as the scaling regularization.

$$\begin{aligned} &\arg\min_{\mathbf{Y},\mathbf{M}} \operatorname{Tr}(\mathbf{Y}\mathbf{F}\mathbf{Y}^\top) + \lambda\textstyle\sum_i D_{KL}\left(\boldsymbol{y}_i \parallel \tilde{\boldsymbol{y}}_i\right) - \gamma\left(\boldsymbol{p}^\top \log \boldsymbol{p}\right) \\ \text{s.t.}\quad &(1-p_i)\boldsymbol{\Omega}\odot\hat{\boldsymbol{y}}_i = \boldsymbol{\Omega}\odot\boldsymbol{y}_i, 0 < p_i < 1, i = l+1, \cdots, n, \mathbf{Y}\mathbf{1}_n = \mathbf{1}_{c+1}, \mathbf{Y} \geq \mathbf{0}. \end{aligned} \tag{16}$$

Here, the graph Laplacian matrix $\mathbf{F}$ is calculated in the feature space. Given a set of examples $\{\boldsymbol{x}_i\}_{i=1}^n$, we can use a k-nearest neighbor graph G to model the relationship

between nearby data points. Specifically, we put an edge between nodes i and j if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are 'close', i.e., $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are among the k nearest neighbors of each other. Define the corresponding weight matrix be $\mathbf{A}$,

$$A_{ij} = \begin{cases} \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_F^2}{\sigma^2}), & \boldsymbol{x}_i \in N_k(\boldsymbol{x}_j) \text{ and } \boldsymbol{x}_j \in N_k(\boldsymbol{x}_i) \\ 0, & \text{otherwise} \end{cases}, \tag{17}$$

where σ is the variance of the Gaussian kernel and is usually estimated according to the average distance between sample points, $N_k(\boldsymbol{x}_i)$ is the set of k neighbors of sample point $\boldsymbol{x}_i$. Then the Laplacian matrix is obtained by $\mathbf{F} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is a diagonal matrix and the diagonal elements are $D_{ii} = \sum_{j=1}^{n} A_{ij}$.

As for optimization, Eq.(16) can be optimized by alternating minimization. In each iteration, we fix one of $\mathbf{Y}, \mathbf{M}$ and update the other. Then the optimization of the original problem can be equivalent to the alternate optimization of the following two sub-problems.

$$\mathbf{M} = \arg\min_{\mathbf{M}} \lambda \sum_i D_{KL}\left(\boldsymbol{y}_i \parallel \tilde{\boldsymbol{y}}_i\right), \tag{18}$$

$$\mathbf{Y} = \arg\min_{\mathbf{Y}} \operatorname{Tr}(\mathbf{Y}\mathbf{F}\mathbf{Y}^\top) + \lambda \sum_i D_{KL}\left(\boldsymbol{y}_i \parallel \tilde{\boldsymbol{y}}_i\right) - \gamma\left(\boldsymbol{p}^\top \log \boldsymbol{p}\right). \tag{19}$$

Here, Eq.(18) can be solved by limited-memory quasi-newton's method (L-BFGS) effectively, which has been used in the existing LDL algorithm SA-BFGS (Geng, 2016). Where $\tilde{\boldsymbol{y}}_i$ is obtained through the maximum entropy model, which has been explained in the subsection 2.2 of the article. It is worth noting that through constraint condition $(1 - p_i)\boldsymbol{\Omega} \odot \hat{\boldsymbol{y}}_i = \boldsymbol{\Omega} \odot \boldsymbol{y}_i$, the updates of $\mathbf{Y}$ and $\boldsymbol{p}$ are synchronized, and the updated $\mathbf{Y}^{t+1}$ can be directly derived from the updated $\boldsymbol{p}^{t+1}$. According to construction of the graph we have

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 = \operatorname{Tr}\left(\boldsymbol{y}^\top \mathbf{F} \boldsymbol{y}\right).$$

Then Eq.(19) can be rewritten as

$$\begin{aligned} \mathbf{Y} &= \arg\min_{\mathbf{Y}} \operatorname{Tr}(\mathbf{Y}\mathbf{F}\mathbf{Y}^\top) + \lambda \sum_i D_{KL}\left(\boldsymbol{y}_i \parallel \tilde{\boldsymbol{y}}_i\right) - \gamma\left(\boldsymbol{p}^\top \log \boldsymbol{p}\right) \\ &= \arg\min_{\mathbf{Y}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 + \lambda \sum_i D_{KL}\left(\boldsymbol{y}_i \parallel \tilde{\boldsymbol{y}}_i\right) - \gamma\left(\boldsymbol{p}^\top \log \boldsymbol{p}\right). \\ &= \arg\min_{\mathbf{Y}} \sum_i \left(\mathbf{A}_{(i)}\mathbf{1}_n \boldsymbol{y}_i{}^\top \boldsymbol{y}_i - \mathbf{A}_{(i)}\mathbf{Y}_{-i}^\top \boldsymbol{y}_i + \lambda \boldsymbol{y}_i^\top \ln \frac{\boldsymbol{y}_i}{\tilde{\boldsymbol{y}}_i} - \gamma p_i \log p_i\right) \end{aligned} \tag{20}$$

where $\mathbf{A}_{(i)}$ represents the i-th row of $\mathbf{A}$, $\mathbf{Y}_{-i}$ represents the label distribution matrix with its i-th column being replaced by a zero vector. It is worth noting that since $\mathbf{A}$ is a symmetric matrix and $A_{ii} = 0$, the cross term

$$\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\boldsymbol{y}_i{}^\top \boldsymbol{y}_j = \sum_i \mathbf{A}_{(i)}\mathbf{Y}_{-i}^\top \boldsymbol{y}_i.$$

Thus, for the optimization of Eq.(19), it can be achieved by alternately optimizing each column $\boldsymbol{y}_i$. It should be noted that due to the symmetry of $\mathbf{A}$, when updating each column, the cross term is calculated twice, so the second term needs to be multiplied by 2 in front.

$$\boldsymbol{y}_i = \arg\min_{\boldsymbol{y}_i} \ \mathbf{A}_{(i)}\mathbf{1}_n {\boldsymbol{y}_i}^\top \boldsymbol{y}_i - 2\mathbf{A}_{(i)}\mathbf{Y}_{-i}^\top \boldsymbol{y}_i + \lambda \boldsymbol{y}_i^\top \ln \frac{\boldsymbol{y}_i}{\tilde{\boldsymbol{y}}_i} - \gamma p_i \log p_i. \tag{21}$$

Substitute the constraint condition $(1-p_i)\boldsymbol{\Omega} \odot \hat{\boldsymbol{y}}_i = \boldsymbol{\Omega} \odot \boldsymbol{y}_i, 1-p_i = r_i$ into Eq.(21), let $\boldsymbol{\Omega} \odot \hat{\boldsymbol{y}}_i = \boldsymbol{d}_i$, then Eq.(21) can be converted to solving for r_i. Merging similar items, we can get the objective function for r_i. Note that $\tilde{\boldsymbol{y}}_i$ is calculated by $\mathbf{M}$ through the maximum entropy model.

$$\begin{aligned} r_i = \arg\min_{r_i} \ & \left(\mathbf{A}_{(i)}\mathbf{1}\right)\left(\boldsymbol{d}_i^\top \boldsymbol{d}_i + 1\right) r_i^2 \\ & + \left[2\mathbf{Y}_{(c+1)}\mathbf{A}_{(i)} - 2\mathbf{A}_{(i)}\mathbf{1} - 2\mathbf{A}_{(i)}\mathbf{Y}_{-i}^\top \boldsymbol{d}_i + \lambda\left(\boldsymbol{d}_i^\top \ln \boldsymbol{d}_i - \boldsymbol{d}_i^\top \mathbf{M}\boldsymbol{x}_i + \mathbf{M}_{(c+1)}\boldsymbol{x}_i\right)\right] r_i \\ & + \lambda r_i \ln r_i + (\lambda - \gamma)(1-r_i)\ln(1-r_i), \end{aligned} \tag{22}$$

where $\mathbf{Y}_{(c+1)}$, $\mathbf{M}_{(c+1)}$ represent the $(c+1)$-th row of $\mathbf{Y}$ and $\mathbf{M}$, respectively. Then, we can use the gradient descent method to update r_i

$$\begin{aligned} \nabla r_i = \ & 2\left(\mathbf{A}_{(i)}\mathbf{1}\right)\left(\boldsymbol{d}_i^\top \boldsymbol{d}_i + 1\right) r_i \\ & + \left[2\mathbf{Y}_{(c+1)}\mathbf{A}_{(i)} - 2\mathbf{A}_{(i)}\mathbf{1} - 2\mathbf{A}_{(i)}\mathbf{Y}_{-i}^\top \boldsymbol{d}_i^\top + \lambda\left(\boldsymbol{d}_i^\top \ln \boldsymbol{d}_i - \boldsymbol{d}_i^\top \mathbf{M}\boldsymbol{x}_i + \mathbf{M}_{(c+1)}\boldsymbol{x}_i\right)\right] \\ & + \lambda(\ln r_i + 1) + (\gamma - \lambda)(\ln(1-r_i) + 1). \end{aligned} \tag{23}$$

In summary, the whole procedures of the proposed **Graph** algorithm are shown in Algorithm 1.

Algorithm 1 Graph

1: Initialize $\mathbf{M}^{(0)}$, $\boldsymbol{p}$, λ and γ;
2: Calculate $\boldsymbol{y}^{(0)}$ with $\boldsymbol{p}$;
3: Calculate $\mathbf{F}$;
4: **while** Stopping criterion is not satisfied **do**
5: Solve $\boldsymbol{p}^{(t+1)}$ by Eq.(22) and equality $1-p_i = r_i$;
6: Calculate $\boldsymbol{y}^{(t+1)}$ with $\boldsymbol{p}^{(t+1)}$;
7: Update $\mathbf{M}^{(t+1)}$ by solving Eq.(18) using L-BFGS;
8: $t = t+1$.
9: **end while**

3.2 Algorithm 2: Manifold

By contrast to Algorithm 1, we build the following model for Algorithm 2.

$$\begin{aligned} \arg\min_{\mathbf{Y},\mathbf{L},\mathbf{M}} \quad & \sum_k \left\|\mathbf{Y}^k - \mathbf{L}^k(\mathbf{L}^k)^\top \mathbf{Y}^k \mathbf{W}^k\right\|_F^2 + \lambda\left\|\mathbf{M}\mathbf{X}^k - \mathbf{Y}^k\right\|_F^2 + \gamma\left\|\mathbf{p}^k\right\|_2^2 \\ \text{s.t.} \quad & \operatorname{diag}(\mathbf{L}^k(\mathbf{L}^k)^\top) = \mathbf{1}, k = 1, 2, \cdots, K, \\ & (1-p_i)\boldsymbol{\Omega} \odot \hat{\boldsymbol{y}}_i = \boldsymbol{\Omega} \odot \boldsymbol{y}_i, 0 < p_i < 1, i = l+1, \cdots, n, \\ & \mathbf{Y}\mathbf{1}_n = \mathbf{1}_{c+1}, \mathbf{Y} \geq \mathbf{0}. \end{aligned} \tag{24}$$

Calling back to the LDCL framework, we will find that $\mathbf{R}^k$ is replaced by $\mathbf{L}^k(\mathbf{L}^k)^\top$ with a constraint $\mathrm{diag}(\mathbf{L}^k(\mathbf{L}^k)^\top) = \mathbf{1}$. This is to avoid the trivial solution and guarantee $\mathbf{R}^k$ to be a normalized Laplacian matrix.

It is worth noting that the Laplacian matrix $\mathbf{R}^k$ is optimized together with the label distribution matrix iteratively, rather than specifying any label correlation matrix, since estimating label correlations from training set with new label directly is noisy. Besides, according to previous studies in (Zhu et al., 2018a), the label correlation matrix may vary among regions, we divide the training set into K regions $\left\{\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^K\right\}$ by k-means clustering, in which $\mathbf{X}^k \in \mathbb{R}^{d\times n_k}$ has n_k instances. Denoted $\hat{\mathbf{Y}}^k$ and $\mathbf{Y}^k$ as the previous observed label distribution matrix and the recovery label distribution matrix corresponding to $\mathbf{X}^k$. Let $\mathbf{R}^k$ be the Laplacian matrix of region k and $\mathbf{W}^k$ are calculated for each region. Intuitively, $\mathbf{W}^k$ characterizes the manifold structure information of the data, and $\mathbf{R}^k$ characterizes the label correlation information. As a result, the manifold enhancement synthesizes the structural information of the feature space and the label correlations.

As to optimization, Eq.(24) can be solved by alternating minimization. We give the updating rules of each variable with others are fixed as follows.

$$\mathbf{L}^k = \arg\min_{\mathbf{L}^k} \ \left\|\mathbf{Y}^k - \mathbf{L}^k(\mathbf{L}^k)^\top\mathbf{Y}^k\mathbf{W}^k\right\|_F^2, \tag{25}$$

$$\mathbf{Y}^k = \arg\min_{\mathbf{Y}^k} \ \left\|\mathbf{Y}^k - \mathbf{L}^k(\mathbf{L}^k)^\top\mathbf{Y}^k\mathbf{W}^k\right\|_F^2 \lambda\left\|\mathbf{M}\mathbf{X}^k - \mathbf{Y}^k\right\|_F^2 + \gamma\left\|\mathbf{p}^k\right\|_2^2, \tag{26}$$

$$\mathbf{M} = \arg\min_{\mathbf{M}} \ \|\mathbf{M}\mathbf{X} - \mathbf{Y}\|_F^2, \tag{27}$$

The gradient descent method can be utilized to optimize Eq.(25). Let

$$\begin{aligned} T(\mathbf{L}^k) &= \left\|\mathbf{Y}^k - \mathbf{L}^k(\mathbf{L}^k)^\top\mathbf{Y}^k\mathbf{W}^k\right\|_F^2 \\ &= \mathrm{Tr}\left[(\mathbf{Y}^k - \mathbf{L}^k(\mathbf{L}^k)^\top\mathbf{Y}^k\mathbf{W}^k)^\top(\mathbf{Y}^k - \mathbf{L}^k(\mathbf{L}^k)^\top\mathbf{Y}^k\mathbf{W}^k)\right]. \end{aligned} \tag{28}$$

Then the gradient of $\mathbf{L}^k$ can be obtained

$$\nabla\mathbf{L}^k = 4(\mathbf{L}^k(\mathbf{L}^k)^\top\mathbf{Y}^k\mathbf{W}^k(\mathbf{W}^k)^\top(\mathbf{Y}^k)^\top - \mathbf{Y}^k(\mathbf{W}^k)^\top(\mathbf{Y}^k)^\top)\mathbf{L}^k. \tag{29}$$

To satisfy the constraint $\mathrm{diag}(\mathbf{L}^k(\mathbf{L}^k)^\top) = \mathbf{1}$, each row of $\mathbf{L}^k$ is projected onto the unit norm ball after each update.

For the optimization of Eq.(26), it can be achieved by alternately optimizing each column $\boldsymbol{y}_i$. First, we rewrite Eq.(26) as follows

$$\begin{aligned} &\mathbf{Y}^k = \arg\min_{\mathbf{Y}^k} \ \left\|\mathbf{Y}^k - \mathbf{L}^k(\mathbf{L}^k)^\top\mathbf{Y}^k\mathbf{W}^k\right\|_F^2 + \lambda\left\|\mathbf{M}\mathbf{X}^k - \mathbf{Y}^k\right\|_F^2 + \gamma\left\|\mathbf{p}^k\right\|_2^2 \\ &= \sum_i \left((\boldsymbol{y}_i^k - \mathbf{L}^k(\mathbf{L}^k)^\top\mathbf{Y}^k\boldsymbol{w}_i^k)^\top(\boldsymbol{y}_i^k - \mathbf{L}^k(\mathbf{L}^k)^\top\mathbf{Y}^k\boldsymbol{w}_i^k) + \lambda(\mathbf{M}\boldsymbol{x}_i^k - \boldsymbol{y}_i^k)^\top(\mathbf{M}\boldsymbol{x}_i^k - \boldsymbol{y}_i^k) + \gamma p_i^{k2}\right). \end{aligned} \tag{30}$$

Then, we fix the $(n-1)$-th column and update the rest column under constraints.

$$\begin{aligned}\boldsymbol{y}_i^k = \arg\min_{\boldsymbol{y}_i^k} \sum_i \Big(&(\boldsymbol{y}_i^k)^\top \boldsymbol{y}_i^k - 2(\boldsymbol{y}_i^k)^\top \mathbf{L}^k (\mathbf{L}^k)^\top \mathbf{Y}^k \boldsymbol{w}_i^k + (\boldsymbol{w}_i^k)^\top (\mathbf{Y}^k)^\top \mathbf{L}^k (\mathbf{L}^k)^\top \mathbf{L}^k (\mathbf{L}^k)^\top \mathbf{Y}^k \boldsymbol{w}_i^k \\ &+ \lambda (\boldsymbol{x}_i^k)^\top (\mathbf{M})^\top \mathbf{M} \boldsymbol{x}_i^k - 2\lambda \boldsymbol{x}_i^{k\top} \mathbf{M}^\top \boldsymbol{y}_i^k + \lambda \boldsymbol{y}_i^{k\top} \boldsymbol{y}_i^k + \gamma (p_i^k)^2 \Big).\end{aligned} \tag{31}$$

Algorithm 2 Manifold

```
Initialize L^(0), M^(0), p^k, λ and γ;
Calculate Y^{k,(0)} with p^k;
Calculate W^k for each region;
while Stopping criterion is not satisfied do
  for k = 1 to K do
    Update L^{k,(t+1)} by Eq.(25);
    Solve p^{k,(t+1)} by Eq.(32) and equality 1 − p_i = r_i;
    Calculate y^{k,(t+1)} with p^{k,(t+1)};
  end for
  y^(t+1) = [y^{1,(t+1)}; y^{2,(t+1)}; ··· ; y^{K,(t+1)}];
  M^(t+1) = (X^⊤X)^{-1}X^⊤y;
  t = t + 1.
end while
```

To simplify the presentation, we omit the superscript k is the derivation of $\mathbf{Y}^k$. Specifically, we analyze $\boldsymbol{y}_i$ and it is easy to find that for $j \neq i$, the second term $\sum_j (\boldsymbol{y}_j)^\top \mathbf{L}\mathbf{L}^\top \mathbf{Y}\boldsymbol{w}_j$ of the above equation still contains $\boldsymbol{y}_i$. Looking back at the construction of $\boldsymbol{w}$, we know that $w_{ii} = 0$. Separate out what is related to $\boldsymbol{y}_i$,

$$\sum_j \boldsymbol{y}_j^\top \mathbf{L}\mathbf{L}^\top \boldsymbol{y}_i \boldsymbol{w}_{ji} = \sum_j \boldsymbol{y}_i^\top \mathbf{L}\mathbf{L}^\top \boldsymbol{y}_j \boldsymbol{w}_{ji} = \boldsymbol{y}_i^\top \mathbf{L}\mathbf{L}^\top \mathbf{Y}_{-i} \boldsymbol{W}_{(i)}^\top.$$

Similarly, for the third term of the above equation,

$$\sum_j (\mathbf{L}\mathbf{L}^\top \mathbf{Y}\boldsymbol{w}_j)^\top (\mathbf{L}\mathbf{L}^\top \mathbf{Y}\boldsymbol{w}_j) = \sum_j \Big((\sum_i \mathbf{L}\mathbf{L}^\top \boldsymbol{y}_i w_{ji})^\top (\sum_i \mathbf{L}\mathbf{L}^\top \boldsymbol{y}_i w_{ji}) \Big).$$

Separate out what is related to $\boldsymbol{y}_i$, for $j \neq i$.

$$\begin{aligned}&\sum_j (\mathbf{L}\mathbf{L}^\top \boldsymbol{y}_i w_{ji}) \top (\mathbf{L}\mathbf{L}^\top \boldsymbol{y}_i w_{ji}) + 2 \sum_j (\mathbf{L}\mathbf{L}^\top \boldsymbol{y}_i w_{ji})^\top (\sum_{k \neq i} (\mathbf{L}\mathbf{L}^\top \boldsymbol{y}_k w_{jk})) \\ &\quad = \mathbf{W}_{(i)} \boldsymbol{W}_{(i)}^\top (\mathbf{L}\mathbf{L}^\top \boldsymbol{y}_i)^\top (\mathbf{L}\mathbf{L}^\top \boldsymbol{y}_i) + 2(\mathbf{L}\mathbf{L}^\top \boldsymbol{y}_i)^\top \mathbf{L}\mathbf{L}^\top \mathbf{Y}_{-i} \mathbf{W} \boldsymbol{W}_{(i)}^\top\end{aligned},$$

where $(\mathbf{L}\mathbf{L}^\top)_{(c+1)}$ represents the $(c+1)$-th row of $\mathbf{L}\mathbf{L}^\top$ and $\mathbf{W}_{(i)}$ is the i-th row of $\mathbf{W}$. $\mathbf{Y}_{-i}$ represents the label distribution matrix with its i-th column being replaced by a zero vector. The remaining similar terms are merged together by pluging the constraint $(1-p_i)\boldsymbol{\Omega} \odot \hat{\boldsymbol{y}}_i =$

$\boldsymbol{\Omega} \odot \boldsymbol{y}_i, 1 - p_i = r_i$ into equation. Let $\boldsymbol{\Omega} \odot \hat{\boldsymbol{y}}_i = \boldsymbol{d}_i$. Eq.(31) can be transformed into an optimization to r_i shown as follows.

$$
\begin{aligned}
r_i = \arg\min_{r_i} \ & \boldsymbol{d}_i{}^\top \boldsymbol{d}_i r_i^2 + (1 - r_i)^2 - 2(\boldsymbol{d}_i \mathbf{L}\mathbf{L}^\top \mathbf{Y}_{-i} \boldsymbol{W}_{(i)}^\top - (\mathbf{L}\mathbf{L}^\top)_{(c+1)} \mathbf{Y}_{-i} \boldsymbol{W}_{(i)}^\top) r_i \\
& - 2(\boldsymbol{d}_i \mathbf{L}\mathbf{L}^\top \mathbf{Y}_{-i} \boldsymbol{w}_i - (\mathbf{L}\mathbf{L}^\top)_{(c+1)} \mathbf{Y}_{-i} \boldsymbol{w}_i) r_i + \mathbf{W}_{(i)} \mathbf{W}_{(i)}^\top \big(\mathbf{L}\mathbf{L}^\top \boldsymbol{d}_i - (\mathbf{L}\mathbf{L}^\top)_{(c+1)}\big)^\top \big(\mathbf{L}\mathbf{L}^\top \boldsymbol{d}_i - (\mathbf{L}\mathbf{L}^\top)_{(c+1)}\big) r_i^2 \\
& + 2\mathbf{W}_{(i)} \mathbf{W}_{(i)}^\top \big((\mathbf{L}\mathbf{L}^\top)_{(c+1)}\big)^\top \big(\mathbf{L}\mathbf{L}^\top \boldsymbol{d}_i - (\mathbf{L}\mathbf{L}^\top)_{(c+1)}\big) r_i + 2\big(\boldsymbol{d}_i{}^\top \mathbf{L}\mathbf{L}^\top \mathbf{L}\mathbf{L}^\top \mathbf{Y}_{-i} \mathbf{W} \boldsymbol{W}_{(i)}^\top \\
& - (\mathbf{L}\mathbf{L}^\top)_{(c+1)} \mathbf{L}\mathbf{L}^\top \mathbf{Y}_{-i} \mathbf{W} \boldsymbol{W}_{(i)}^\top\big) r_i + \lambda\big(\boldsymbol{d}_i^\top \boldsymbol{d}_i r_i^2 + (1 - r_i)^2\big) - 2\lambda\big(\boldsymbol{x}_i{}^\top \mathbf{M}^\top \boldsymbol{d}_i - \boldsymbol{x}_i{}^\top \mathbf{M}_{(c+1)}^\top\big) r_i + \gamma(1 - r_i)^2 \\
= \arg\min_{r_i} \ & \Big(\mathbf{W}_{(i)} (\mathbf{W}_{(i)})^\top (\mathbf{L}\mathbf{L}^\top \boldsymbol{d}_i - (\mathbf{L}\mathbf{L}^\top)_{(c+1)})^\top (\mathbf{L}\mathbf{L}^\top \boldsymbol{d}_i - (\mathbf{L}\mathbf{L}^\top)_{(c+1)}) + (1 + \lambda)(\boldsymbol{d}_i^\top \boldsymbol{d}_i + 1) + \gamma\Big) r_i^2 \\
& + 2\Big(- (\boldsymbol{d}_i \mathbf{L}\mathbf{L}^\top \mathbf{Y}_{-i} \boldsymbol{W}_{(i)}^\top - (\mathbf{L}\mathbf{L}^\top)_{(c+1)} \mathbf{Y}_{-i} \boldsymbol{W}_{(i)}^\top) - (\boldsymbol{d}_i \mathbf{L}\mathbf{L}^\top \mathbf{Y}_{-i} \boldsymbol{w}_i - (\mathbf{L}\mathbf{L}^\top)_{(c+1)} \mathbf{Y}_{-i} \boldsymbol{w}_i) \\
& \quad + \mathbf{W}_{(i)} \mathbf{W}_{(i)}^\top \big((\mathbf{L}\mathbf{L}^\top)_{(c+1)}\big)^\top \big(\mathbf{L}\mathbf{L}^\top \boldsymbol{d}_i - (\mathbf{L}\mathbf{L}^\top)_{(c+1)}\big) + \boldsymbol{d}_i{}^\top \mathbf{L}\mathbf{L}^\top \mathbf{L}\mathbf{L}^\top \mathbf{Y}_{-i} \mathbf{W} \boldsymbol{W}_{(i)}^\top \\
& \quad - (\mathbf{L}\mathbf{L}^\top)_{(c+1)} \mathbf{L}\mathbf{L}^\top \mathbf{Y}_{-i} \mathbf{W} \boldsymbol{W}_{(i)}^\top - \lambda\big(\boldsymbol{x}_i{}^\top \mathbf{M}^\top \boldsymbol{d}_i - \boldsymbol{x}_i{}^\top \mathbf{M}_{(c+1)}^\top\big) - 1 - \gamma - \lambda\Big) r_i.
\end{aligned}
\tag{32}
$$

Through observation, it can be found that Eq.(32) is a quadratic equation of one variable r_i, which has a closed-form solution through the extreme value formula.

The optimal $\mathbf{M}$ can be derived by least square method directly, which has the closed form solution $\mathbf{M} = \mathbf{Y}\mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top)^{-1}$.

In summary, the whole procedures of the proposed **Manifold** algorithm are shown in Algorithm 2.

3.3 Theoretical Analysis

For the two algorithms mentioned above, we have further conducted generalization ability and convergence analysis, and then given the following corollaries and conclusion.

As mentioned in the LDCL framework, for the maximum entropy model output, the softmax function $\boldsymbol{\phi}$ is used for normalization

$$
\tilde{y}_{ij} = p(y_{ij} | \boldsymbol{x}_i; \mathbf{M}) = \frac{1}{S_i} \exp\left(\sum_r M_{jr} x_{ir}\right),
$$

where $S_i = \sum_j \exp(\sum_r M_{jr} x_{ir})$ is a normalization factor. Actually, the maximum entropy model can be regarded as a combination of softmax function and multi-output linear regression, namely $\boldsymbol{\phi} \circ \mathcal{H}$, where $\mathcal{H}$ represents a class of functions of multi-output linear regression.

Graph model uses KL divergence as loss function, which is denoted by $KL : \mathbb{R}^{(c+1)} \times \mathbb{R}^{(c+1)} \to \mathbb{R}_+$. Rademacher complexity of **Graph** w.r.t. $\mathbf{Z}$ for loss function KL satisfies the following lemma.

Lemma 6 *Let $\mathcal{H}$ be a family of functions for multi-output linear regression, Rademacher complexity of **Graph** with KL loss satisfies*

$$
\Re_n(KL \circ \boldsymbol{\phi} \circ \mathcal{H} \circ \mathbf{Z}) \leq \sqrt{2}(\sqrt{c+1} + 1)(c+1) \Re_n(\mathcal{H} \circ \mathbf{Z}). \tag{33}
$$

Proof Note that $KL(\boldsymbol{\mu}, \cdot)$ is not ρ-Lipschitz over $\mathbb{R}^{(c+1)}$ for any $\rho \in \mathbb{R}$ and $\boldsymbol{\mu} \in \mathbb{R}^{(c+1)}$, thus Support-Theorem 4 cannot be applied directly. Next we show that $KL \circ \boldsymbol{\phi}(\boldsymbol{\mu}, \cdot)$ satisfies Lipschitzness. For any $\mathbf{p}, \mathbf{q} \in \mathbb{R}^{(c+1)}$,

$$\begin{aligned} |KL \circ \boldsymbol{\phi}(\boldsymbol{\mu}, \mathbf{p}) - KL \circ \boldsymbol{\phi}(\boldsymbol{\mu}, \mathbf{q})| &= \left| \sum_{i=1}^{c+1} \mu_i \left(\ln \frac{\exp(p_i)}{\sum_{i=1}^{c+1} \exp(p_i)} - \ln \frac{\exp(q_i)}{\sum_{i=1}^{c+1} \exp(q_i)} \right) \right| \\ &\leq \sum_{i=1}^{c+1} \mu_i \left| \ln \left(1 + \sum_{j \neq i} e^{p_j - p_i} \right) - \ln \left(1 + \sum_{j \neq i} e^{q_j - q_i} \right) \right|, \end{aligned} \tag{34}$$

where p_i, q_i is i-th element of $\mathbf{p}$ and $\mathbf{q}$, respectively. Observing that $\ln \left(1 + \sum_{j \neq i} e^{\nu_i} \right)$ is 1-Lipschitz for $\boldsymbol{\nu} \in \mathbb{R}^{(c+1)}$, thus right-hand side of preceding equation is bounded by

$$\sum_{i=1}^{c+1} \mu_i \|\mathbf{p} - \mathbf{1}p_i - \mathbf{q} + \mathbf{1}q_i\|_2 \leq \|\mathbf{p} - \mathbf{q}\|_2 + \sqrt{c+1} \sum_{i=1}^{c+1} \mu_i |p_i - q_i| \leq (\sqrt{c+1} + 1) \|\mathbf{p} - \mathbf{q}\|_2, \tag{35}$$

where the last inequality is according to Cauchy-Schwarz inequality. Thus, $KL \circ \boldsymbol{\phi}(\boldsymbol{\mu}, \cdot)$ is $(\sqrt{c+1} + 1)$-Lipschitz. Then according to Support-Theorem 4, lemma 6 is proved. ■

Similarly, we can get

$$\Re_{l+u}^{Td}(KL \circ \boldsymbol{\phi} \circ \mathcal{F} \circ \mathbf{Z}) \leq \sqrt{2}(\sqrt{c+1} + 1)(c+1) \Re_{l+u}^{Td}(\mathcal{F} \circ \mathbf{Z}).$$

Manifold model uses L_2-norm as loss function, which is denoted by $L_2 : \mathbb{R}^{(c+1)} \times \mathbb{R}^{(c+1)} \to \mathbb{R}_+$. Rademacher complexity of **Manifold** w.r.t. $\mathbf{Z}$ for loss function L_2 satisfies the following lemma.

Lemma 7 *Let $\mathcal{H}$ be a family of functions for multi-output linear regression, Rademacher complexity of **Manifold** with L_2 loss satisfies*

$$\Re_n(L_2 \circ \boldsymbol{\phi} \circ \mathcal{H} \circ \mathbf{Z}) \leq 2\sqrt{2}(c+1)^2 \Re_n(\mathcal{H} \circ \mathbf{Z}). \tag{36}$$

Proof

$$|L_2 \circ \boldsymbol{\phi}(\mathbf{p}, \cdot) - L_2 \circ \boldsymbol{\phi}(\mathbf{q}, , \cdot)| = \sum_{i=1}^{c+1} \left| \frac{1}{1 + \sum_{j \neq i} e^{p_j - p_i}} - \frac{1}{1 + \sum_{j \neq i} e^{q_j - q_i}} \right|.$$

Observing that $\left(1 + \sum_i e^{\nu_i} \right)$ is 1-Lipschitz for $\boldsymbol{\nu} \in \mathbb{R}^{(c+1)}$, thus the preceding equation is bounded by

$$\begin{aligned} &\sum_{i=1}^{c+1} \|\mathbf{p} - 1p_i - \mathbf{q} + 1q_i\|_2 \leq \sum_{i=1}^{c+1} \left(\|\mathbf{p} - \mathbf{q}\|_2 + \sqrt{c+1} \, |p_i - q_i| \right) \\ &\leq (c+1) \|\mathbf{p} - \mathbf{q}\|_2 + \sqrt{c+1} \sum_{i=1}^{c+1} |p_i - q_i| \leq 2(c+1) \|\mathbf{p} - \mathbf{q}\|_2 \quad . \end{aligned} \tag{37}$$

Thus, $L_2 \circ \boldsymbol{\phi}$ is $2(c+1)$-Lipschitz. Then according to Support-Theorem 4, lemma 7 is proved. ■

Similarly, we can get

$$\Re_{l+u}^{Td}(L_2 \circ \boldsymbol{\phi} \circ \mathcal{F} \circ \mathbf{Z}) \leq 2\sqrt{2}(c+1)^2 \Re_{l+u}^{Td}(\mathcal{F} \circ \mathbf{Z}).$$

Based on Theorem 4, Theorem 5, lemma 6 and lemma 7, we can get the following corollaries about **Graph** Algorithm and **Manifold** Algorithm.

Corollary 8 *Denote softmax function by $\boldsymbol{\phi}$ as the normalization, Let $\mathcal{F}$ be the family of transductive mapping functions for **Graph** defined above with KL divergence as loss function bounded by B. Given $\mathbf{Z} = \mathbf{Z}_l \cup \mathbf{Z}_u$ as a the full sample. Define $Q \triangleq \frac{l+u}{(l+u-1/2)(1-1/(2\max(l,u)))}$ and $P \triangleq \frac{lu}{(l+u)^2}$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$ the following bound holds for all $f \in \mathcal{F}$,*

$$\begin{aligned} \hat{err}_u(f) \leq \sqrt{2}(\sqrt{c+1}+1)(c+1)\Re_{l+u}^{Td}(\mathcal{F} \circ \mathbf{Z}) + B\left(\frac{l+u}{lu}\right)\sqrt{\frac{32\ln(4e)}{3}}\sqrt{\min(l,u)} \\ + B\sqrt{\frac{Q}{2}\left(\frac{l+u}{lu}\right)\ln\left(\frac{1}{\delta}\right)}. \end{aligned} \tag{38}$$

For the second item,

Corollary 9 *Denote softmax function by $\boldsymbol{\phi}$ as the normalization, Let $\mathcal{H}$ be the family of inductive mapping functions for **Graph** defined above with KL divergence as loss function bounded by B. Given $\mathbf{Z} = \mathbf{Z}_l \cup \mathbf{Z}_u$ as a the full sample, such that $n = l + u$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$ the following bound holds for all $h \in \mathcal{H}$,*

$$\begin{aligned} err_{(\boldsymbol{x},\boldsymbol{y})\sim\boldsymbol{\mathcal{D}}}(h) \leq \hat{err}_{l+u}(h) + \hat{err}_u(f) + 2\sqrt{2}(\sqrt{c+1}+1)(c+1)\Re_n(\mathcal{H} \circ \mathbf{Z}) \\ + \frac{2\sqrt{2}(\sqrt{c+1}+1)(c+1)u}{l+u}\Re_{l+u}^{Td}(L_2 \circ \mathcal{F} \circ \mathbf{Z}) + B\sqrt{\frac{\log(\frac{1}{\sigma})}{2(l+u)}}. \end{aligned} \tag{39}$$

Corollary 10 *Denote softmax function by $\boldsymbol{\phi}$ as the normalization. Let $\mathcal{F}$ be the family of transductive mapping functions for **Manifold** defined above with L_2 loss as loss function bounded by a constant $\sqrt{2}$. Given $\mathbf{Z} = \mathbf{Z}_l \cup \mathbf{Z}_u$ as a the full sample. Define $Q \triangleq \frac{l+u}{(l+u-1/2)(1-1/(2\max(l,u)))}$ and $P \triangleq \frac{lu}{(l+u)^2}$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$ the following bound holds for all $f \in \mathcal{F}$,*

$$\begin{aligned} \hat{err}_u(f) \leq 2\sqrt{2}(c+1)^2\Re_{l+u}^{Td}(\mathcal{F} \circ \mathbf{Z}) + \sqrt{2}\left(\frac{l+u}{lu}\right)\sqrt{\frac{32\ln(4e)}{3}}\sqrt{\min(l,u)} \\ + \sqrt{Q\left(\frac{l+u}{lu}\right)\ln\left(\frac{1}{\delta}\right)}. \end{aligned} \tag{40}$$

Corollary 11 *Denote softmax function by $\boldsymbol{\phi}$ as the normalization, Let $\mathcal{H}$ be the family of inductive mapping functions for **Manifold** defined above with L_2 loss as loss function*

bounded by $\sqrt{2}$. *Given* $\mathbf{Z} = \mathbf{Z}_l \cup \mathbf{Z}_u$ *as a the full sample, such that* $n = l + u$. *Then for any* $\delta \in (0, 1)$, *with probability at least* $1 - \delta$ *the following bound holds for all* $h \in \mathcal{H}$,

$$\begin{aligned} err_{(\boldsymbol{x},\boldsymbol{y})\sim\boldsymbol{\mathcal{D}}}(h) \leq \hat{err}_{l+u}(h) + \hat{err}_u(f) + 2\sqrt{2}(c+1)^2\Re_n(\mathcal{H}\circ\mathbf{Z}) \\ + \frac{2\sqrt{2}(c+1)^2 u}{l+u}\Re^{Td}_{l+u}(\mathcal{F}\circ\mathbf{Z}) + \sqrt{\frac{\log(\frac{1}{\sigma})}{l+u}} \quad . \end{aligned} \tag{41}$$

From Corollary 8 to Corollary 11, we can find that the transductive risk bound of transductive mapping is tighter than the generalization bound of inductive mapping, which is consistent with the conclusions of Theorem 4 and 5 under the LDCL framework. In particular, for the two specific algorithms, using different loss functions ℓ when designing the model will lead to different generalization bounds. In addition, different regularization terms will also constrain different hypothesis function spaces to correspond to different Rademacher complexity.

Theorem 12 *According to the procedures in Algorithm 1 and Algorithm 2, after updating a set of variables* $\boldsymbol{y}$, $\boldsymbol{p}$ *and* $\mathbf{M}$ *in each iteration, the objective function* $\mathcal{J}(\boldsymbol{y}; \mathbf{M}; \boldsymbol{p})$ *of the model is non-increasing, and finally our algorithm converge to a local optimal solution. For any* $\varepsilon > 0$, *there is a number* N, *so that when the number of iterations* $t > N$, *the following inequalities holds*

$$\begin{aligned} &\left\| \mathcal{J}^{(t+1)}(\mathbf{Y}^{(t+1)}; \mathbf{M}^{(t+1)}; \boldsymbol{p}^{(t+1)}) - \mathcal{J}^{(t)}(\boldsymbol{y}^{(t)}; \mathbf{M}^{(t)}; \boldsymbol{p}^{(t)}) \right\| \leq \varepsilon \\ &\mathcal{J}^{(t+1)}(\mathbf{Y}^{(t+1)}; \mathbf{M}^{(t+1)}; \boldsymbol{p}^{(t+1)}) \leq \mathcal{J}^{(t)}(\boldsymbol{y}^{(t)}; \mathbf{M}^{(t)}; \boldsymbol{p}^{(t)}). \end{aligned} \tag{42}$$

Proof First of all, according to the LDCL framework, we denote the objective function as follows

$$\mathcal{J}(\boldsymbol{y}; \mathbf{M}; \mathbf{p}) = \mathcal{L}(\mathbf{Y}) + \lambda\mathcal{H}(\mathbf{M}, \mathbf{Y}) + \gamma_1\mathcal{R}(\boldsymbol{p}). \tag{43}$$

The alternative optimization criterion is adopted, and the optimal solutions of variables $\boldsymbol{y}$, $\mathbf{M}$ and $\boldsymbol{p}$ are obtained after each fixed update. In the iteration $t+1$, the following inequality holds

$$\boldsymbol{p}^{(t+1)} = \underset{0\leq m_i\leq 1}{\arg\min}\, \mathcal{L}(\mathbf{Y}^{(t)}) + \lambda\mathcal{H}(\mathbf{M}^{(t)}, \mathbf{Y}^{(t)}) + \gamma\mathcal{R}(\boldsymbol{p}^{(t)}), \tag{44}$$

$$\boldsymbol{y}^{(t+1)} = \underset{\boldsymbol{y}\times\mathbf{1_n}=\mathbf{1_{(c+1)}}}{\arg\min}\, \mathcal{L}(\mathbf{Y}^{(t)}) + \lambda\mathcal{H}(\mathbf{M}^{(t)}, \mathbf{Y}^{(t)}), \tag{45}$$

$$\mathbf{M}^{(t+1)} = \arg\min \mathcal{H}(\mathbf{M}^{(t)}, \mathbf{Y}^{(t+1)}). \tag{46}$$

It is worth noting that through the constraint condition $(1 - p_i)\boldsymbol{\Omega} \odot \hat{\boldsymbol{y}}_i = \boldsymbol{\Omega} \odot \boldsymbol{y}_i$, the solution of Eq.(45) can be directly calculated from the solution of Eq.(44), which makes the optimization process more concise. According to the alternating optimization criterion of Eq.(44,45,46), we can understand that the objective function is no-increase monotonically with the increase of iteration times, that is, the following inequality is satisfied.

$$\mathcal{J}^{(t+1)}(\mathbf{Y}^{(t+1)}; \mathbf{M}^{(t+1)}; \boldsymbol{p}^{(t+1)}) \leq \mathcal{J}^{(t)}(\boldsymbol{y}^{(t)}; \mathbf{M}^{(t)}; \boldsymbol{p}^{(t)}). \tag{47}$$

On the other hand, through the constraint condition $(1-p_i)\mathbf{\Omega}\odot\hat{\boldsymbol{y}}_i=\mathbf{\Omega}\odot\boldsymbol{y}_i$, our solution has been carried out in the feasible region, and the convergence satisfies the KKT condition of the optimization problem. Since each item of the objective function is greater than 0, the objective function is non-negative. According to the principle of monotonic bounds, we can ensure that the objective function converges to a locally optimal solution. For any $\varepsilon>0$, there is a number N, so that when the number of iterations $t>N$, the following inequalities holds

$$\left\|\mathcal{J}^{(t+1)}(\mathbf{Y}^{(t+1)};\mathbf{M}^{(t+1)};\boldsymbol{p}^{(t+1)})-\mathcal{J}^{(t)}(\boldsymbol{y}^{(t)};\mathbf{M}^{(t)};\boldsymbol{p}^{(t)})\right\|\leq\varepsilon. \tag{48}$$

■

From Theorem 12, we can know that both the **Graph** and **Manifold** algorithms will converge to a stable value after a finite number of iterations. Furthermore, we will conduct a convergence experiment in the following Subsection 5.5 to demonstrate this property.

4. Related Work

Label ambiguity comes from label information scarcity. With the evolution of data collection ways, incomplete, inexact and inaccurate label information may cause label ambiguity (Zhou, 2018; Li et al., 2021). Among different types of label ambiguity, LDL (Geng, 2016), which expresses label ambiguity by giving each instance a label distribution, and ENC (Park and Shim, 2010), which focuses on model reusing with new classes, are two typical cases and there are a lot related researches (Gao et al., 2016; Mu et al., 2016).

LDL is a novel machine learning framework for dealing with label ambiguity, which assumes that the labels are related to each instance to some degree and gives each instance a label distribution. LDL was first proposed to solve the problem of facial age estimation (Geng et al., 2010), since the label distribution matches the continuity of age changes. Later, Geng (Geng et al., 2013) finds that in many practical applications, the distribution across all labels was better than the correlation between a single label and an instance. Since LDL can provide more ambiguous label information, it has been successfully applied in many real scenarios in recent years, such as facial age estimation (Geng et al., 2010), facial expression recognition (Zhou et al., 2015), text mining (Zhou et al., 2016) and so on.

The existing LDL algorithms are mainly divided into three categories, i.e., problem transformation, algorithm adaptive methods and specific algorithms. Compared with our framework and algorithms, they can not be utilized to solve our problem directly since they focus on the traditional well-defined LDL setting, without specification about the label distribution changing. Our designed algorithms are related to these traditional methods since they share the same components as shown in Table 1.

In addition, some other weak-supervision within the LDL paradigm is also considered. For example, Xu et al. (Xu and Zhou, 2017) proposed an label incomplete LDL (IncomLDL) method based on trace norm minimization (Cai et al., 2010), which considers the label correlation via low rank assumption (Xu et al., 2013). Jia et.al. (Jia et al., 2019) proposed a weakly supervised label distribution learning algorithm based on transductive matrix completion and label correlation. Xu et al. (Xu et al., 2021a) proposed a novel inductive fragmentary LDL algorithm via graph regularized maximum entropy criteria (GRME), which

explores the correlation between labels based on graph regularization matrix reconstruction, together with a classifier for categorization. These works differ from our work since they focus on the incompleteness of labels while we focus on the emerging of new class.

Another closely-related topic is ENC. Typical ENC problems include Multi-label learning with emerging new labels (MuENL) (Zhu et al., 2018b), which mainly solves the problem of new classes in multi-label learning, including the recognition and prediction of new classes; classification under streaming emerging new classes (SENC) (Mu et al., 2016), which is used to solve the classification problem with emerging new classes in the multi-class problem in streaming data; and Multi-Instance learning with Emerging Novel class (MIEN) (Wei et al., 2021), which focuses on the emerging novel class problem in multi-instance learning. Obviously, due to the differences in data formats and learning paradigms, the existing ENC will not be applicable to LDCL. Similarly, some incremental learning methods that focus on multi-class problems, such as (Cermelli et al., 2020; Rebuffi et al., 2017), will not be able to adapt to the special scene of LDCL due to the unique label format of LDL.

It is also worth noting that in multi-label learning, which assumes the feature space is a continuous Euclidean space and the instances are distributed on a low-dimensional manifold, and the label space is a discrete logical space. Different from that, both the feature space and the label space of our work are continuous Euclidean space. Thus, traditional multi-label algorithms are not suitable for our settings either.

5. Experiment

In this section, we will compare the proposed methods with the adapted LDL methods in two experimental settings.

5.1 Data Sets and Evaluation Measures

We evaluate our methods and comparative methods on the real-world data sets. There are in total 13 data sets including biology, movie ratings, emotional analysis and so on. The Yeast series datasets are real-world datasets collected from biological experiments with Saccharomyces cerevisiae. For each dataset, the number of labels represents discrete time points in a biological experiment. The gene expression level at each time point naturally gives the corresponding label description degree, and then the value of the gene expression level at each time point is normalized to form a label distribution. The Natural Scene dataset is derived from 2000 natural scenes, and the 9 possible labels associated with these images are plants, sky, clouds, snow, buildings, deserts, mountains, water, and sun. These images are then independently annotated by an annotator, Then, the inconsistent rankings for each image are transformed into a label distribution by a nonlinear programming process. The Movie dataset is about user ratings of movies, according to the percentage calculated for each movie's rating label distribution. The brief statistics of these data sets are shown in Table 2. More details of them can be found in the literature (Geng, 2016).

There are totally five different metrics in LDL (Geng, 2016). These measures can be divided into two groups. The first group, i.e., Chebyshev, Clark and Canberra, measure the distance between the two distributions. The lower the values of these metrics, the better the performance of the algorithms("↓"). The second group, i.e., Cosine and Intersection, measure the similarity between the two distributions. The higher the values of these metrics, the

Table 2: Statistics of the 13 data sets, where n is the number of instance, d is the number of features and c is the number of labels.

Dataset	n	d	c
Yeast-alpha	2465	24	18
Yeast-cdc	2465	24	15
Yeast-elu	2465	24	14
Yeast-diau	2465	24	7
Yeast-heat	2465	24	6
Yeast-spo	2465	24	6
Yeast-cold	2465	24	4
Yeast-dtt	2465	24	4
Yeast-spo5	2465	24	3
Yeast-spoem	2465	24	2
emotion6	1980	168	7
Natural Scene	2000	294	9
MovieDataSet	7755	1869	5

better the performance of the algorithms("↑"). We take Chebyshev and Intersection as two representatives. Note that there is one additional measurement proposed in (Geng, 2016), which measures the KL-divergence between two vectors. KL-divergence is calculated by $\boldsymbol{y}_i^\top \log\left(\boldsymbol{y}_i/\tilde{\boldsymbol{y}}_i\right)$, and it will be meaningless when $\tilde{\boldsymbol{y}}_i$ is zero, we will not use it here.

5.2 Settings and Baselines

In order to obtain the label distribution data accompanying the emerging new label, we transform the existing label distribution data. First select the new label, we averaged each class label value and chose the label class with the smallest label value as the new label. This also conforms to the prior assumption that the new label information is a supplementary to the existing label information. Without loss of generality, we divide the data into two parts, with 10% of the data as the test data and 90% as the training data. The training data is divided into two parts, 10% of the training data is relabeled data with emerging new labels, and 90% of the training data is un-relabeled data. On the other hand, for the un-relabeled training data, we remove the corresponding new labels and then normalize the remaining labels to meet the label distribution. In this way, we get experimental data that meets the background of the question in this article. We conduct experiments on two settings to verify the effectiveness of our method.

Setting 1: We verify the accuracy of reconstructing the expanded label distribution matrix of the training data, that is, the recovery ability for the expanded label distribution matrix of un-relabeled data.

Setting 2: We compare the performance of the proposed method on the test set by measuring the difference between the ground-truth and the predicted label distribution matrix, that is, the prediction ability for the new coming data.

Table 3: Chebyshev ("↓")(mean±std) results in setting 1 on un-relabeled data. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
Yeast-alpha	.0074±.0006	**.0036±.0001**	.0142±.0002	.0156±.0009	.3673±.0254	.0201±.0002
Yeast-cdc	.0121±.0007	**.0067±.0006**	.0174±.0001	.0186±.0008	.3869±.0268	.0235±.0002
Yeast-elu	.0153±.0020	**.0077±.0004**	.0173±.0002	.0201±.0020	.3832±.0231	.0241±.0002
Yeast-diau	.0315±.0006	**.0286±.0011**	.0392±.0007	.0459±.0027	.4158±.0304	.0456±.0003
Yeast-heat	.0345±.0005	**.0251±.0022**	.0447±.0007	.0476±.0026	.4302±.0570	.0528±.0004
Yeast-spo	.0430±.0012	**.0304±.0059**	.0617±.0008	.0653±.0014	.4180±.0239	.0658±.0004
Yeast-cold	**.0353±.0003**	.0371±.0029	.0537±.0005	.0580±.0028	.4272±.0348	.0614±.0004
Yeast-dtt	.0237±.0005	**.0220±.0015**	.0384±.0004	.0421±.0030	.4222±.0265	.0497±.0004
Yeast-spo5	**.0628±.0005**	.0814±.0094	.0976±.0015	.0994±.0054	.4082±.0307	.0979±.0003
Yeast-spoem	.0922±.0004	**.0884±.0010**	.0919±.0018	.0965±.0070	.3229±.0247	.0930±.0008
emotion6	.1532±.0019	**.0399±.0004**	.3746±.0993	.3507±.0125	.3493±.0011	.3259±.0029
Natural scene	**.0312±.0009**	.0526±.0009	.6740±.0230	.4156±.0179	.3681±.0024	.3581±.0030
MovieDataSet	.0784±.0017	**.0772±.0005**	.1559±.0036	.2426±.0221	.1806±.0001	.1600±.0013
win/tie/loss	4/0/9	9/0/4	0/0/13	0/0/13	0/0/13	0/0/13

In the experiments, the parameters λ and γ are both selected by grid searching from $\{10^{-4}, 10^{-3}, \cdots, 10^{4}\}$ by cross-validation on training data. For Algorithm 2, we add a clustering parameter k. Similarly, we use the grid search method to select the best number of clusters from $[1, 2, \cdots, 9]$ through cross-validation on training data. The maximum iteration is set to be 100. The stopping criterion parameter ϵ is set to be 10^{-3}.

We compare our proposed LDCL algorithms with several baselines. The representative LDL algorithms include two maximum entropy algorithms IIS-LDL, BFGS-LDL (Geng, 2016) and two problem transformation algorithms PT-Bayes and PT-SVM (Geng, 2016). Therefore, we adopt these four methods as baselines. All the codes are shared by original authors, and we use the suggested default parameters. However, these baseline methods can not be utilized in our setting, we adapt existing label distribution learning algorithms directly and naively to fit the situation of this article. On one hand, we only use a small amount of relabeled data with new labels to train traditional models to learn a classifier and measure the difference between the ground-truth and the predicted label distribution matrix on the un-relabeled data of training set. On the other hand, we can relabel a large amount of un-relabeled data through completion and normalize it into distribution. Here, we use the mean filling strategy, that is, the new labels for the unrelabeled data are filled with the mean of the new labels for the relabeled data and then renormalized to obtain the expanded label distribution matrix. They are then used as training data to train the traditional model and give a prediction for test data. These two kinds of data adaptation correspond to the two sets of comparative experiments in the article.

Table 4: Intersection ("↑")(mean±std) results for setting 1 on un-relabeled data. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
Yeast-alpha	.9926±.0004	**.9964±.0001**	.9597±.0002	.9535±.0025	.4420±.0326	.9430±.0003
Yeast-cdc	.9630±.0008	**.9933±.0006**	.9536±.0003	.9505±.0017	.4449±.0206	.9392±.0002
Yeast-elu	.9643±.0020	**.9923±.0004**	.9561±.0004	.9478±.0054	.4521±.0229	.9404±.0004
Yeast-diau	.9685±.0006	**.9714±.0011**	.9368±.0011	.9260±.0045	.4960±.0239	.9254±.0004
Yeast-heat	.9655±.0005	**.9749±.0022**	.9368±.0010	.9318±.0044	.5057±.0500	.9238±.0005
Yeast-spo	.9570±.0012	**.9696±.0059**	.9108±.0012	.9056±.0024	.5125±.0198	.9045±.0006
Yeast-cold	**.9647±.0003**	.9629±.0029	.9378±.0006	.9331±.0028	.5415±.0379	.9290±.0004
Yeast-dtt	.9763±.0005	**.9780±.0015**	.9557±.0004	.9515±.0035	.5554±.0246	.9424±.0004
Yeast-spo5	**.9372±.0005**	.9186±.0094	.9024±.0015	.9006±.0054	.5918±.0307	.9021±.0003
Yeast-spoem	.9078±.004	**.9116±.0010**	.9081±.0018	.9035±.0070	.6771±.0247	.9070±.0008
emotion6	.8468±.0019	**.9601±.0004**	.5068±.0825	.5124±.0239	.5275±.0010	.5594±.0031
Natural scene	**.9688±.0010**	.9474±.0010	.2832±.0215	.3626±.0323	.3630±.0016	.4620±.0038
MovieDataSet	.9215±.0017	**.9228±.0013**	.7838±.0046	.6738±.0434	.7395±.0002	.7863±.0017
win/tie/loss	3/0/10	10/0/3	0/0/13	0/0/13	0/0/13	0/0/13

Table 5: Chebyshev ("↓")(mean±std) results for setting 2 on test set. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
Yeast-alpha	.0141±.0002	**.0140±.0000**	.0141±.0001	.0146±.0004	.1135±.0080	.0213±.0000
Yeast-cdc	.0164±.0007	**.0156±.0001**	.0168±.0001	.0171±.0006	.1113±.0072	.0231±.0001
Yeast-elu	.0163±.0005	**.0158±.0001**	.0161±.0001	.0168±.0003	.1147±.0069	.0234±.0001
Yeast-diau	.0466±.0004	**.0384±.0012**	.0387±.0004	.0438±.0042	.1508±.0120	.0458±.0003
Yeast-heat	.0521±.0001	**.0436±.0001**	.0467±.0001	.0477±.0011	.1754±.0097	.0550±.0005
Yeast-spo	.0702±.0004	**.0583±.0001**	.0592±.0006	.0641±.0038	.1911±.0062	.0661±.0005
Yeast-cold	.0650±.0007	**.0559±.0003**	.0672±.0015	.0765±.0089	.1912±.0210	.0763±.0013
Yeast-dtt	.0465±.0001	**.0365±.0001**	.0529±.0015	.0571±.0065	.1776±.0184	.0608±.0013
Yeast-spo5	.0938±.0001	**.0851±.0005**	.1099±.0035	.1132±.0071	.2222±.0191	.1164±.0032
Yeast-spoem	.0876±.0012	**.0867±.0006**	.1717±.0048	.1830±.0180	.2162±.0133	.1743±.0048
emotion6	.3343±.0003	**.3072±.0001**	.3262±.0001	.3580±.0188	.6718±.0110	.3232±.0006
Natural scene	**.3472±.0003**	.3546±.0006	.3639±.0102	.4342±.0258	.4116±.0025	.3672±.0001
MovieDataSet	**.1337±.0018**	.1698±.0007	.1435±.0004	.2300±.0243	.1988±.0030	.1530±.0001
win/tie/loss	2/0/11	11/0/2	0/0/13	0/0/13	0/0/13	0/0/13

Table 6: Intersection ("↑")(mean±std) results for setting 2 on test set. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
Yeast-alpha	.9607±.0002	**.9614±.0000**	.9610±.0001	.9584±.0007	.7467±.0061	.9402±.0001
Yeast-cdc	.9578±.0003	**.9586±.0002**	.9585±.0002	.9551±.0017	.7636±.0076	.9401±.0001
Yeast-elu	.9562±.0004	**.9581±.0002**	.9571±.0001	.9544±.0011	.7719±.0099	.9413±.0001
Yeast-diau	.9244±.0003	**.9389±.0009**	.9383±.0003	.9261±.0072	.7829±.0129	.9249±.0003
Yeast-heat	.9238±.0002	**.9375±.0000**	.9335±.0007	.9318±.0019	.7666±.0105	.9202±.0007
Yeast-spo	.8990±.0005	**.9150±.0002**	.9131±.0009	.9075±.0038	.7496±.0050	.9023±.0007
Yeast-cold	.9250±.0007	**.9368±.0003**	.9250±.0014	.9137±.0099	.7875±.0227	.9137±.0012
Yeast-dtt	.9456±.0003	**.9575±.0004**	.9404±.0015	.9353±.0060	.8011±.0175	.9308±.0013
Yeast-spo5	.9062±.0001	**.9149±.0005**	.8901±.0035	.8868±.0071	.7778±.0191	.8836±.0032
Yeast-spoem	.9124±.00012	**.9133±.0007**	.8283±.0048	.8170±.0180	.7838±.0133	.8257±.0048
emotion6	.5402±.0005	**.5753±.0004**	.5535±.0002	.5152±.0251	.2885±.0080	.5629±.00010
Natural scene	**.4827±.0010**	.4804±.0011	.4745±.0417	.3506±.0466	.3506±.0013	.4590±.0013
MovieDataSet	**.8187±.0009**	.7540±.0012	.8021±.0004	.6963±.0425	.7234±.0025	.7963±.0002
win/tie/loss	2/0/11	11/0/2	0/0/13	0/0/13	0/0/13	0/0/13

5.3 Classification Accuracy

Due to space limitations, here we only present representative results Chebyshev (the lower the better) and Intersection (the higher the better). Other results are similar, and we report them in the Appendix D.3. As mentioned above, we conduct two sets of experiments to verify the recovery ability of the proposed methods for the un-relabeled data and the predictive ability for the new testing data. In each setting, we conduct 20 runs and report the mean±std. In the first set of experiments, we compare the performance of the proposed models to the traditional LDL algorithms on unlabeled data. The results are shown in Tables 3 and 4. In the second set of experiments, we compare the prediction performance of the proposed models to the traditional LDL algorithms on new coming data. The results are shown in Tables 5 and 6. For the traditional LDL algorithms, we first complete the emerging new label of the un-relabeled data by mean filling strategy, and then normalize it into a distribution to train the models. The best results on each measure are marked in bold. In addition, we have also investigated the significance between our method and other methods by t-test at 95% significance level.

As we can see from the experimental results, our methods have achieved the best performance in almost all cases, which demonstrates the superiority of our model. In particular, as seen from the results in Tables 3 and 4 in setting 1, the accuracy of reconstructed the expanded label distribution matrix of the training data, that is, the recovery ability for the un-relabeled data achieved a high level. Moreover, comparing the performance of the traditional LDL algorithms in the two sets of experiments, we find that completing the emerging new labels of the un-relabeled data can alleviate the lack of training data. Nevertheless, it will also introduce noise and make the model performance worse at the same time. On the

other hand, comparing the two specific algorithms under the model framework, Manifold enhancement methods achieve better performance than Graph methods on most datasets. On the rest datasets, the two algorithms have achieved comparable results for recovery ability, but show differences in terms of predictive ability. It indicates that the proposed model framework has a certain degree of inclusiveness. The model combination within the framework also has different characteristics and shows different performance for different datasets.

5.4 Ablation experiments

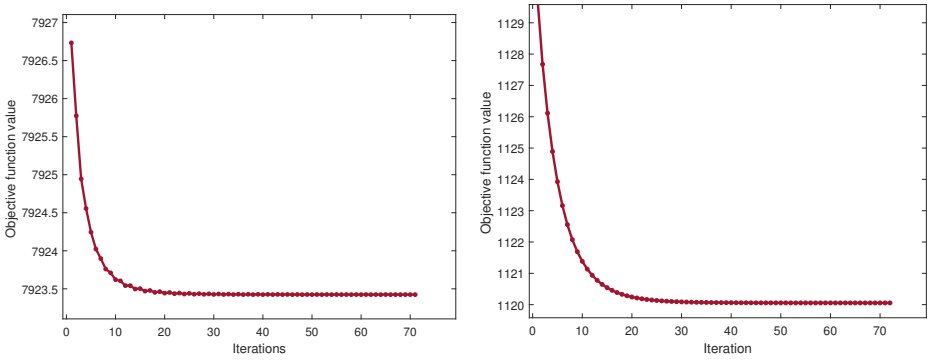
In order to demonstrate the effectiveness of various parts of the framework, we conduct ablation experiments to illustrate the contribution of different components in LDCL.

- **Without Scaling Regularization (SR)** we only consider the empirical risks for calculating the expanded label distribution matrix and learning a classifier by setting $\lambda = 0$.
- **Without Classifier Learning (CL)** in this setting, we set $\gamma = 0$ and only consider the empirical risks for calculating the expanded label distribution matrix under the scaling Regularization.

The results of the ablation experiment are shown in Table 7 and Table 8. It should be pointed out that the ablation experiments are carried out under setting 1, and the main comparison is the recovery ability for the expanded label distribution matrix of un-relabeled data. From the results, we can draw two conclusions. First of all, compared with the complete model, removing any one of the two terms will lead to performance degeneration, which show that each term does contribute to the performance improvement of LDCL. Secondly, among these two terms, the scaling regularization plays a more important role. This proves that the main purpose of the introduction of the second term is to learn an inductive hypothesis, which is consistent with the original design intention of the LDCL model.

Table 7: Chebyshev (the lower the better)(mean±std) results and Intersection (the higher the better)(mean±std) results of ablation experiments. The best results on each row are bold. (pairwise single-tailed t-test at 95% confidence level)

Metrics	Chebyshev			Intersection		
Dataset	Graph	SR	CL	Graph	SR	CL
Yeast-heat	**.0275±.0005**	.0361±.0013	.0300±.0025	**.9725±.0005**	.9639±.0013	.9700±.0025
Yeast-alpha	**.0074±.0006**	.0442±.0037	.0175±.0030	**.9926±.0004**	.9558±.0037	.9825±.0030
Yeast-spo	**.0304±.0059**	.0499±.0020	.0457±.0036	**.9570±.0012**	.9501±.0020	.9543±.0036
Yeast-cdc	**.0121±.0007**	.0423±.0023	.0378±.0029	**.9630±.0008**	.9577±.0023	.9622±.0029
Natural scene	**.0312±.0009**	.0511±.0012	.0352±.0006	**.9688±.0010**	.8997±.0012	.9588±.0006
MovieDataSet	**.0784±.0017**	.1002±.0006	.0829±.0014	**.9215±.0017**	.8997±.0006	.9171±.0014
win/tie/loss	6/0/0	0/0/6	0/0/6	6/0/0	0/0/6	0/0/6



(a) Algorithm1: Graph (b) Algorithm2: Manifold

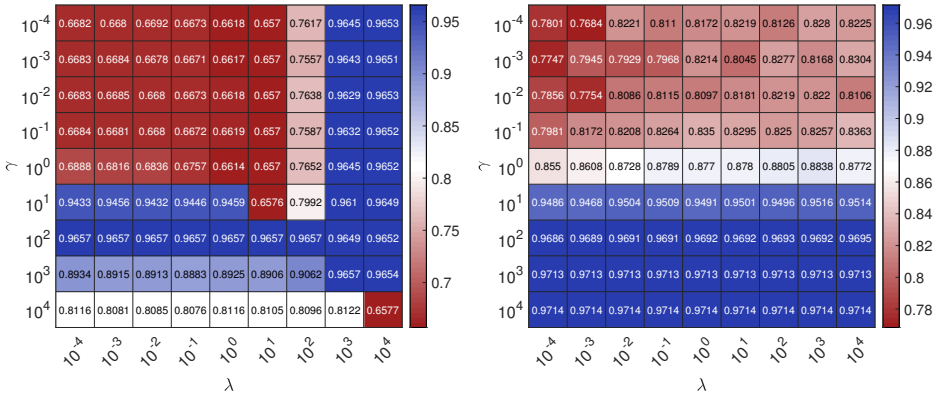
Figure 4: The convergence curves of the proposed algorithms on Yeast-heat. The first one is the convergence curves of Algorithm 1 and the latter one is the convergence curves of Algorithm 2.

Table 8: Chebyshev (the lower the better)(mean±std) results and Intersection (the higher the better)(mean±std) results of ablation experiments. The best results on each row are bold. (pairwise single-tailed t-test at 95% confidence level)

Metrics	Chebyshev			Intersection		
Dataset	Manifold	SR	CL	Manifold	SR	CL
Yeast-heat	**.0251±.0022**	.0976±.0189	.0343±.0037	**.9749±.0022**	.9024±.0189	.9657±.0037
Yeast-alpha	**.0036±.0001**	.0514±.0012	.0059±.0004	**.9964±.0001**	.9486±.0012	.9941±.0004
Yeast-spo	**.0430±.0012**	.0941±.0157	.0469±.0011	**.9696±.0059**	.9059±.0157	.9531±.0011
Yeast-cdc	**.0067±.0006**	.0426±.0074	.0087±.0010	**.9933±.0006**	.9574±.0074	.9913±.0010
Natural scene	**.0526±.0009**	.1745±.0093	.0613±.0006	**.9474±.0010**	.8255±.0093	.9068±.0006
MovieDataSet	**.0772±.0005**	.1471±.0314	.0883±.0059	**.9228±.0013**	.8529±.0314	.9117±.0059
win/tie/loss	6/0/0	0/0/6	0/0/6	6/0/0	0/0/6	0/0/6

5.5 Convergence Analysis

As shown in Theorem 12, our methods will converge to a local minima. In order to verify the convergence of the proposed algorithms, we present the curves of objective function values of the two algorithms on the Yeast-heat data sets. The results on other date sets are the same. From the convergence curves of these two data sets shown in Fig.4, it can be seen that when the number of iterations increases, the value of the objective function does not increase and gradually converges to a fixed value. In addition, the objective function decreases rapidly in the first few iterations and converges in no more than 30 iterations. It indicates that our methods have a fast convergence rate. This may be caused by the characteristics of the gradient method and the quasi-Newton method, which are mainly used in alternating minimization. For these two methods, the convergence rate is affected by the size of the gradient. As the value of the objective function approaches the minimum value, the gradient



(a) Algorithm1: Graph (b) Algorithm2: Manifold

Figure 5: The influence of the parameters on the performance of the two algorithms under Intersection metric on the Natural scene. Note that for Algorithm 2, we fix the maximum number of clustering regions $K = 9$ and k is set as $k = 2$.

decreases. Therefore, the value of the objective function decreases rapidly in the first few iterations, and then slowly decreases until convergence.

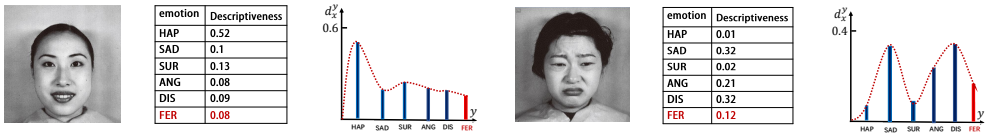
5.6 Parameter Analysis

To get the best parameters of the proposed algorithms, we conduct 5-fold cross validation on the training set. In Algorithm 1, there are two parameters λ and γ. The sensitivity of these two parameters on the Natural scene is shown in the left plane of Fig.5. It is worth noting that for criteria Intersection, the larger the value is, the better the performance is. In Algorithm 2, there are three parameters k, λ and γ. For visualization, we fix $k = 2$ and the sensitivity of other two parameters on the Natural scene is shown in the right plane of Fig.5.

It can be seen from the Fig.5 that the parameters of Algorithm 1 are more sensitive than those of Algorithm 2. This may be due to the more complex classifiers and regularization terms in Algorithm 1. Then, the corresponding parameters have a greater impact on optimization. For Algorithm 2, we find that the parameter λ has a small effect on the performance of the algorithm, which may be due to the stable performance of the least squares fitting. Furthermore, as the parameter γ increases within a certain range, the performance of Algorithm 2 increases gradually, and then tends to be stable. It can validate the effectiveness of the regularization term.

5.7 Emotion Distribution Recognition

In practical applications, people's facial expressions are often complex and diverse. An expression rarely expresses pure emotion, but often a mixture of different emotions. In addition, the degree of each emotion describes an expression becomes a question worthy of attention. Emotion Distribution Recognition from facial expressions (Zhou et al., 2015) is playing an increasingly important role in autonomous driving and criminal interrogation or psychological counseling. However, with the continuous advancement of human cognition and the developmental innovation of technology, people are pursuing more detailed insight



(a) (b)

Figure 6: Two typical expressions from SJAFFE and SBU_3DFE respectively. The 'Fear' marked in red is a new facial expression, and the right is the corresponding emotion distribution predicted by the proposed methods.

and detection of facial expressions. Although existing methods have achieved satisfactory results in the distribution of expression recognition problem, the exiting LDL can not handle the Emotion Distribution Changing Recognition problem due to the emerging of new emotions.

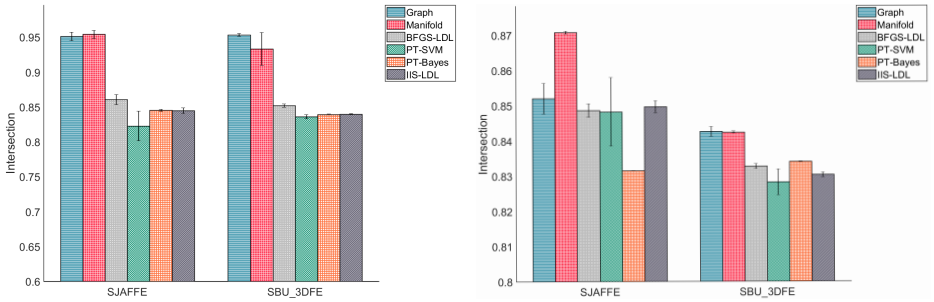
We extend two widely used facial expression databases: SJAFFE and SBU_3DFE (Yin et al., 2006), to the emotion distribution case. The JAFFE database contains 213 gray scale expression images. And the 243-dimensional feature vector was extracted from each image by local binary mode (LBP) (Zhou et al., 2015). Each image was rated by 60 people on a five-point scale on five emotional labels (i.e., happiness, sadness, surprise, anger and disgust). The average score for each emotion is used to indicate emotional intensity. We modify this set to SJAFFE to fit our setting. It not only considers the emotions with the highest scores, as in most of the work on JAFFE, but also retains all the scores and normalizes them into a label distribution for all five emotion labels. Similarly, the BU_3DFE contains 2,500 facial expression images, each of them is scored by 23 students in the same way as JAFFE. The specific scores on each basic emotion are obtained and transferred into emotion distributions.

In the labeling process, it was observed that these facial expressions also contain fear to a certain extent. It corresponds to the emerging new emotion label. Since relabeling the data will cost a lot, we apply the proposed LDCL model to deal with the Emotion Distribution Changing Recognition problem Fig.6 shows two examples from SJAFFE and SBU_3DFE respectively. A small amount of data is relabeled by the marked red *fear*, and we use a small amount of relabeled data and a large amount of un-relabeled data to design algorithms to recover the expended label distribution matrix of un-relabeled data and make predictions for new coming data. The proposed methods are compared with the existing LDL algorithms, and the experimental results are shown in Table 9 and Fig.7. Concretely, Table 9 shows the recovery ability (corresponds to setting 1) of the proposed method for un-relabeled data, and the prediction ability (corresponds to setting 2) for new coming data under Chebyshev metric. Fig.7 shows the the recovery ability and prediction ability of the proposed method under Intersection metric.

It can be found that our methods have achieved satisfactory results. Significantly, in setting 1, our methods have achieved great advantages compared to the other methods, since both the un-relabeled data and the re-labeled data participate in the procedures in the training of the proposed methods. In setting 2, for the prediction ability of new coming data, Algorithm 2 performs better than Algorithm 1. As seen from the variance of Chebyshev results, the possible reason is that the classifier of Algorithm 2 is more stable, which can also be mutually confirmed with the previous parameter analysis results.

Table 9: Chebyshev (the lower the better)(mean±std) results of two Algorithms under two experiment settings. The best results on each row are bolded. (pairwise single-tailed t-test at 95% confidence level)

Setting	Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
setting1	SJAFFE	.0488±.0060	**.0458±.0058**	.1103±.0062	.1318±.0116	.1227±.0014	.1233±.0017
	SBU_3DFE	**.0465±.0012**	.0667±.0235	.1175±.0023	.1408±.0032	.1382±.0005	.1340±.0007
setting2	SJAFFE	.1029±.0009	**.0994±.0003**	.1275±.0006	.1279±.0053	.1546±.0001	.1228±.0003
	SBU_3DFE	.1202±.0004	**.1198±.0003**	.1321±.0003	.1422±.0039	.1382±.0008	.1325±.0007



(a) Setting1 (b) Setting2

Figure 7: Intersection (the higher the better) (mean and std.) results of two Algorithms on data sets SJAFFE and SBU_3DFE under two experiment settings.

6. Conclusion

In this paper, we formulate a new framework to solve the problem of Label Distribution Changing Learning (LDCL), which is brand new and of great importance. It expands the sample space by rescaling the previous distribution and further mining the topological information of the sample space to restore the emerging new class. Specifically, the formulated LDCL framework consists of three parts. The first part is based on graph learning. Label Propagation (LP) and Manifold Learning (ML) are used to reconstruct the expanded label distribution matrix. The second part trains a classifier for categorization. In the third part, we design the scaling regularization for the constraints from three perspectives. The integration of these three terms will facilitate the model to get better performance. Moreover, the corresponding generalization error bounds of the LDCL framework are derived to support the model framework. In this paper, we only focus on adding one type of emerging label. How to extend it into multiple types of labels is an interesting future work. A possible way may be that we can add them one by one. Besides, how to accelerate the optimization speed is also worth studying. Several modern optimization tools should be utilized for alleviating computational burden.

Acknowledgments

This work was partially supported by the National Key Research and Development Program (No.2018YFB1305101), the Key NSF of China under Grant No. 62136005, 62036013, the NSF of China under Grant No. 61922087. Chenping Hou and Dewen Hu are the corresponding authors.

Appendix A. Preliminaries

In this section, we introduce basic notations and definitions for the following generalization error bound analysis.

A.1 Learning Setup

According to the scenario of this article, the data type meets the transduction setting. Therefore, we define a full sample set $\mathbf{Z} = \{Z_1, ..., Z_{l+u}\}$ consisting of arbitrary $l+u$ points from $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is feature space and $\mathcal{Y}$ is label space, each $\boldsymbol{y} \in \mathcal{Y}$ is a label distribution matrix, and each column $\boldsymbol{y} \in \boldsymbol{y}$ is a label distribution. Then the full sample set $\mathbf{Z}$ is divided into $\mathbf{Z}_l$ and $\mathbf{Z}_u$. Noting that $\mathbf{Z}_l$ is chosen uniformly at random without replacement from $\mathbf{Z}$, and the instances in which are annotated by the new label distribution from the expended label sample space. The instances in the remaining sample set $\mathbf{Z}_u = \mathbf{Z}_{l+u} \backslash \mathbf{Z}_l$ are annotated by the previous label distribution called un-relabeled set. According to the proposed LDCL framework, both $\mathbf{Z}_l$ and $\mathbf{Z}_u$ participate in training, and then we obtain a transductive mapping $f \in \mathcal{F}$ and an inductive mapping $h \in \mathcal{H}$, where $\mathcal{F}$ and $\mathcal{H}$ represent the transductive and inductive mappings hypothesis space of scoring functions $f : \mathbf{Z}_l \times \mathbf{Z}_u \to \boldsymbol{y}_{l+u}$ and $h : \mathcal{X} \to \mathcal{Y}$. The quality of f and h is measured by $err_u(f)$ and $err_{(x,y)\sim\mathcal{D}}(h)$ which have been defined in the main text, where $\mathcal{D}$ is the data distribution of the sample $(\boldsymbol{x}, \boldsymbol{y})$ in $\mathcal{X} \times \mathcal{Y}$.

A.2 Supplementary Definitions and Support-Theorems

In this part, we will introduce some necessary definitions and Support-Theorems, some simple definitions mentioned in the main text are omitted.

Definition 13 (Random Permutation Vector (Chen and Cheng, 1999; Stanley, 2007)) *Let $\mathbf{Z} \triangleq \mathbf{Z}_1^{l+u} \triangleq (Z_1, ..., Z_{m_1+m_2})$ be a random permutation vector where the variable Z_k, $k \in \mathbf{I}_1^{m_1+m_2}$ is the k-th component of a permutation of $\mathbf{I}_1^{m_1+m_2}$ that is chosen uniformly at random. Let $\mathbf{Z}_{ij}$ be a perturbed permutation vector obtained by exchanging the positions of Z_i and Z_j in $\mathbf{Z}$.*

Any function f on permutations of $\mathbf{I}_1^{m_1+m_2}$ is called (m_1, m_2)-permutation symmetric if $f(\mathbf{Z}) \triangleq f(Z_1, ..., Z_{m_1+m_2})$ is symmetric on $Z_1, ..., Z_{m_1}$ as well as on $Z_{m_1+1}, ..., Z_{m_1+m_2}$.

Definition 14 (Pairwise Rademacher variables (El-Yaniv and Pechyony, 2009)) *Let $\hat{\boldsymbol{\sigma}} = \{\sigma_i\}_{i=1}^{m_1+m_2}$ be a vector of i.i.d. random variables defined as*

$$\hat{\boldsymbol{\sigma}}_i = (\hat{\sigma}_{i,1}, \hat{\sigma}_{i,2}) = \begin{cases} \left(-\frac{1}{m_1}, -\frac{1}{m_2}\right) & with\ Prob.\ \frac{m_1 m_2}{(m_1+m_2)^2}; \\ \left(-\frac{1}{m_1}, \frac{1}{m_1}\right) & with\ Prob.\ \frac{{m_1}^2}{(m_1+m_2)^2}; \\ \left(\frac{1}{m_2}, \frac{1}{m_1}\right) & with\ Prob.\ \frac{m_1 m_2}{(m_1+m_2)^2}; \\ \left(\frac{1}{m_2}, \frac{1}{m_2}\right) & with\ Prob.\ \frac{{m_2}^2}{(m_1+m_2)^2}; \end{cases} . \tag{49}$$

The Definition 14 is derived from Definition 3 (with $P_0 = \frac{m_1 m_2}{(m_1+m_2)^2}$) in the following way. When the Rademacher variable $\sigma_i = 1$, corresponding to $(\hat{\sigma}_{i,1}, \hat{\sigma}_{i,2}) = (\frac{1}{m_2}, \frac{1}{m_1})$, $\sigma_i = -1$ corresponds to $(\hat{\sigma}_{i,1}, \hat{\sigma}_{i,2}) = (-\frac{1}{m_1}, -\frac{1}{m_2})$, and $\sigma_i = 0$ then we split it at random to $\left(-\frac{1}{m_1}, \frac{1}{m_1}\right)$ or $\left(\frac{1}{m_2}, -\frac{1}{m_2}\right)$.

The proofs of Theorem 4 and Theorem 5 are standard, and we provide in here to make the appendix self-contained. To prove these two theorems, we need following concentration inequalities and Support-Theorems.

Support-Theorem 1 (Concentration Inequalities (El-Yaniv and Pechyony, 2006)) *Based on de Definition 13, let* $\mathbf{Z}$ *be a random permutation vector over* $\mathbf{I}_1^{m_1+m_2}$ *and* $f(\mathbf{Z})$ *be an* (m_1, m_2)*-permutation symmetric function satisfying* $\left|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})\right| \leq \beta$ *for all* $i \in \mathbf{I}_1^{m_1}$, $j \in \mathbf{I}_{l+1}^{m_1+m_2}$. *Then the following probability inequality holds*

$$P_{\mathbf{Z}}\{f(\mathbf{Z}) - E_{\mathbf{Z}}\{f(\mathbf{Z})\} \geq \varepsilon\} \leq \exp\left(-\frac{2\varepsilon^2(m_1+m_2-1/2)}{m_1 m_2 \beta^2}\left(1 - \frac{1}{2\max(m_1, m_2)}\right)\right). \quad (50)$$

The proof of Support-Theorem 1 relies on McDiarmid's inequality (McDiarmid, 1989, Corollary 6.10) (Mcdiarmid, 1989).

Remark 15 *The inequality Eq.(50) is defined for any* (m_1, m_2)*-permutation symmetric function* f. *By specializing* f, *we can obtain the following concentration inequality. Let* $g : \mathbf{I}_1^{m_1+m_2} \to [0, B]$ *and*

$$f(\mathbf{Z}) = \frac{1}{m_2}\sum\nolimits_{i=m_1+1}^{m_1+m_2} g(Z_i) - \frac{1}{m_1}\sum\nolimits_{i=1}^{m_1} g(Z_i),$$

then $E_{\mathbf{Z}}\{f(\mathbf{Z})\} = 0$. *Moreover, for any* $i \in \mathbf{I}_1^{m_1}$, $j \in \mathbf{I}_{m_1+1}^{m_1+m_2}$, $\left|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})\right| \leq B(\frac{1}{m_1} + \frac{1}{m_2})$. *Therefore, by specializing Support-Theorem 1 for such* f *we have*

$$\begin{aligned} &P_{\mathbf{Z}}\{\frac{1}{m_2}\sum_{i=l+1}^{m_1+m_2} g(Z_i) - \frac{1}{m_1}\sum_{i=1}^{l} g(Z_i)\} \geq \varepsilon\} \\ &\leq \exp\left(-\frac{\varepsilon^2 m_1 m_2 (m_1+m_2-1/2)}{B^2(m_1+m_2)^2} \cdot \frac{2\max(m_1, m_2) - 1}{\max(m_1, m_2)}\right) \end{aligned}. \quad (51)$$

Support-Theorem 2 (Uniform Concentration Inequality (El-Yaniv and Pechyony, 2009)) *Let* $\mathcal{V}$ *be a set of vectors in* $[B_1, B_2]^{m_1+m_2}$, $B_1 \leq 0$, $B_2 \geq 0$ *and set* $B \triangleq B_2 - B_1$, $B_{\max} = \max(|B_1| - |B_2|)$. *Consider two independent permutations of* $\mathbf{I}_1^{m_1+m_2}$, $\mathbf{Z}$ *and* $\mathbf{Z}'$. *For any* $\boldsymbol{v} \in \mathcal{V}$ *denoted by*

$$\boldsymbol{v}(\mathbf{Z}) \triangleq (\boldsymbol{v}(Z_1), \boldsymbol{v}(Z_2), ..., \boldsymbol{v}(Z_{m_1+m_2})),$$

the vector $\boldsymbol{v}$ *permuted according to* $\mathbf{Z}$. *We use the following abbreviations for averages of* $\boldsymbol{v}$ *over subsets of its components:* $\mathbf{H}_k\{\boldsymbol{v}(\mathbf{Z})\} \triangleq \frac{1}{m_1}\sum_{i=1}^{k} \boldsymbol{v}(Z_i)$, $T_k\{\boldsymbol{v}(\mathbf{Z})\} \triangleq \frac{1}{m_2}\sum_{i=k+1}^{m_1+m_2} v(Z_i)$ *(note that* H *stands for 'head' and* T *for 'tail'). In the special case where* $k = m_1$ *we set*

$H\{\boldsymbol{v}(\mathbf{Z})\} \triangleq H_{m_1}\{\boldsymbol{v}(\mathbf{Z})\}$, *and* $T\{\boldsymbol{v}(\mathbf{Z})\} \triangleq T_{m_1}\{\boldsymbol{v}(\mathbf{Z})\}$. *The uniform concentration inequality that we develop shortly states that for any* $\delta > 0$, *with probability at least* $1-\delta$ *over random permutation* $\mathbf{Z}$ *of* $\mathbf{I}_1^{m_1+m_2}$, *for any* $\boldsymbol{v} \in \mathcal{V}$, *we can obtain*

$$T\{\boldsymbol{v}(\mathbf{Z})\} \leq H\{\boldsymbol{v}(\mathbf{Z})\} + \Re_{m_1+m_2}(\mathcal{V}) + o\left(\sqrt{\frac{1}{\min(m_1,m_2)}\ln\frac{1}{\delta}}\right). \tag{52}$$

Support-Theorem 3 (Bernstein-type concentration inequality (Devroye et al., 1996)) *For the binomial random variable* $\mathbf{Z} \sim \boldsymbol{\mathcal{B}}(n,p)$. *We have the following probability inequality*

$$P_{\mathbf{Z}}\{|\mathbf{Z} - E\mathbf{Z}| > \varepsilon\} < 2\exp\left(-\frac{3\varepsilon^2}{8np}\right). \tag{53}$$

Support-Theorem 4 (Rademacher Vector Contraction Inequality (Maurer, 2016)) *Let* $\mathcal{T}$ *be a class of real functions, and* $\mathcal{H} \subset \mathcal{T} = \mathcal{T}_1 \times ... \times \mathcal{T}_C$ *be a* C*-valued function class. If* $\Phi : \mathbf{R}^C \mapsto \mathbf{R}$ *is a* L*-Lipschitz continuous function and* $\Phi(0) = 0$, *then* $\hat{\Re}_{\mathbf{Z}}(\Phi \circ H) \leq \sqrt{2}L\sum_{i=1}^{C}\hat{\Re}_{\mathbf{Z}}(\mathcal{T}_i)$.

Support-Theorem 5 (McDiarmids inequality (Zhao and Zhou, 2018)) *Define a set of independent random variables* $Z_1, Z_2, \cdots, Z_n \in \boldsymbol{\mathcal{Z}}$ *and assume that there exist a set of* $B_i > 0$. *if* $f : \mathbf{Z}^n \to \mathbb{R}$ *is a real-valued function, and for any* $\mathbf{Z}^n \in \boldsymbol{\mathcal{Z}}$ *satisfies the following condition*

$$\left|f(Z_1, \cdots, Z_i, \cdots, Z_n) - f\left(Z_1, \cdots, Z_i', \cdots, Z_n\right)\right| \leq B_i.$$

for all $i \in [1;n]$ *and any point* $Z_1, Z_2, \cdots, Z_n, Z'_i \in \boldsymbol{\mathcal{Z}}$. *To simplify the presentation, let* $f(\mathbf{Z}) = f(Z_1, \cdots, Z_i', \cdots, Z_n)$, *then for any* $\varepsilon > 0$, *the following inequalities hold*

$$\begin{aligned} \Pr[f(\mathbf{Z}) - Ef(\mathbf{Z}) \geq \varepsilon] &\leq \exp\left(\tfrac{-2\varepsilon^2}{\sum_{i=1}^{n} B_i^2}\right) \\ \Pr[f(\mathbf{Z}) - Ef(\mathbf{Z}) \leq -\varepsilon] &\leq \exp\left(\tfrac{-2\varepsilon^2}{\sum_{i=1}^{n} B_i^2}\right) \end{aligned}. \tag{54}$$

Appendix B. Proof of Theorem 4

Proof To prove Theorem 4. We need to introduce random permutation vector and permutation symmetric function as defined in Definition 13. Let $\mathcal{L} \circ \mathcal{F}$ corresponds the family of loss functions ℓ associated to the output function space $\mathcal{F}$. For any $f \in \mathcal{F}$, we donate $\mathcal{L} \circ f = (\ell \circ f(Z_1), \cdots, \ell \circ f(Z_{l+u}))$ as a loss vector on the sample $\mathbf{Z}$. Consider two samples $\mathbf{Z}_{l+u}$ and $\mathbf{Z}'_{l+u}$ of size $l+u$. According to Definition 13, we know that $\mathbf{Z}$ and $\mathbf{Z}'$ are two independent permutations of $\mathbf{I}_1^{l+u}$. By comparison, it can be found that the definitions of $H\{\boldsymbol{v}(\mathbf{Z})\}$, $T\{\boldsymbol{v}(\mathbf{Z})\}$ in Support-Theorem 2 and $\hat{err}_u(f)$, $\hat{err}_l(f)$ are equivalent. Then we can obtain the following inequality based on Support-Theorem 2.

$$\hat{err}_u(f) \leq \hat{err}_l(f) + \Re_{l+u}^{Td}(\mathcal{L} \circ \mathcal{F}) + o\left(\sqrt{\frac{1}{\min(l,u)}\ln\frac{1}{\delta}}\right). \tag{55}$$

Then we will further analyze Eq.(55) in detail. We denote $\bar{\mathcal{L}} \circ f \triangleq \frac{1}{l+u}\sum_{i=1}^{l+u} \ell \circ f(Z_i)$ as the average component of $\mathcal{L} \circ f$. For any $f \in \mathcal{F}$ and any sample $\mathbf{Z}$ of size $l+u$, we can get

$$\begin{aligned}
\hat{err}_u(f) &= \hat{err}_l(f) + \hat{err}_u(f) - \hat{err}_l(f) \\
&\leq \hat{err}_l(f) + \sup_{f\in\mathcal{F}}\left[\hat{err}_u(f) - \bar{\mathcal{L}} \circ f + \bar{\mathcal{L}} \circ f - \hat{err}_l(f)\right] \\
&= \hat{err}_l(f) + \sup_{f\in\mathcal{F}}[\hat{err}_u(f) - E_{\mathbf{Z}'}\hat{err}_u(f_{\mathbf{Z}'}) + E_{\mathbf{Z}'}\hat{err}_l(f_{\mathbf{Z}'}) - \hat{err}_l(f)]\cdot \\
&\leq \hat{err}_l(f) + E_{\mathbf{Z}'}\sup_{f\in\mathcal{F}}[\hat{err}_u(f) - \hat{err}_u(f_{\mathbf{Z}'}) + \hat{err}_l(f_{\mathbf{Z}'}) - \hat{err}_l(f)]
\end{aligned} \tag{56}$$

For convenience, we let

$$\varphi(\mathbf{Z}) \triangleq E_{\mathbf{Z}'}\sup_{f\in\mathcal{F}}[\hat{err}_u(f) - \hat{err}_u(f_{\mathbf{Z}'}) + \hat{err}_l(f_{\mathbf{Z}'}) - \hat{err}_l(f)],$$

$$Q \triangleq \frac{l+u}{(l+u-1/2)(1-1/(2\max(l,u)))}.$$

For sufficiently large l and u, the value of Q is almost 1. The function $\varphi(\mathbf{Z})$ is (l,u)-permutation symmetric in $\mathbf{Z}$. For a loss function ℓ bounded by B, it can be verified that $\left|\varphi(\mathbf{Z}) - \varphi(\mathbf{Z}^{ij})\right| \leq B\left(\frac{1}{l}+\frac{1}{u}\right)$. Therefore, we can apply Support-Theorem 1 and Remark 15 with $\beta \triangleq B\left(\frac{1}{l}+\frac{1}{u}\right)$ to $\varphi(\mathbf{Z})$. Since $\hat{err}_u(f) - \hat{err}_l(f) \leq \varphi(\mathbf{Z})$. We obtain, with probability of at least $1-\delta$ over random permutation $\mathbf{Z}$ of $\mathbf{I}_1^{l+u}$, for all $f \in \mathcal{F}$:

$$\hat{err}_u(f) \leq \hat{err}_l(f) + E_{\mathbf{Z}}\left\{\varphi(\mathbf{Z})\right\} + B\sqrt{\frac{Q}{2}\left(\frac{1}{l}+\frac{1}{u}\right)\ln\frac{1}{\delta}}. \tag{57}$$

In order to analyze $\varphi(\mathbf{Z})$, we need introduce some new notions.

- **Step1:** $R_{l+u}^{Td}(\mathcal{L} \circ \mathcal{F} \circ \mathbf{Z}) = E_{N_1,N_2}\boldsymbol{s}(N_1,N_2)$.
- **Step2:** $E_{\mathbf{Z}}\left\{\varphi(\mathbf{Z})\right\} = \boldsymbol{s}(E_{\hat{\boldsymbol{\sigma}}}N_1, E_{\hat{\boldsymbol{\sigma}}}N_2)$.
- **Step3:** Apply Support-Theorem 3 to get

$$E_{N_1,N_2}\left|\boldsymbol{s}(N_1,N_2) - \boldsymbol{s}(E_{\hat{\boldsymbol{\sigma}}}N_1, E_{\hat{\boldsymbol{\sigma}}}N_2)\right| \leq B\sqrt{32\ln(4e)/3}\left(\frac{1}{l}+\frac{1}{u}\right)\sqrt{u}.$$

Based on Pairwise Rademacher variables, it is easy to verify that

$$\hat{\Re}_{l+u}^{Td}(\mathcal{L} \circ \mathcal{F} \circ \mathbf{Z}) = E_{\hat{\boldsymbol{\sigma}}}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{(l+u)}\left(\sigma_{i,1}+\sigma_{i,2}\right)\ell \circ f(Z_i)\right]. \tag{58}$$

According to Definition 14, we know that $\hat{\boldsymbol{\sigma}}_{\boldsymbol{i}}$ is a discrete random variable. Let n_1, n_2, and n_3 correspond to the number of random variables $\hat{\boldsymbol{\sigma}}_i$ realizing the value $(\frac{1}{u},\frac{1}{l})$, $(-\frac{1}{l},\frac{1}{l})$ (or $(\frac{1}{u},-\frac{1}{u})$) and $(-\frac{1}{l},-\frac{1}{u})$, respectively. Denote $N_1 = n_1+n_2$ and $N_2 = n_3+n_2$, we know that both N_i and n_i are random variables. Then we further define the probability distribution of

$\hat{\boldsymbol{\sigma}}_{\boldsymbol{i}}$ conditioned on the events $n_1+n_2=N_1$ and $n_2+n_3=N_2$, denoted as $\mathcal{C}(N_1,N_2)$. Based on the constrained distribution $\mathcal{C}(N_1,N_2)$, we let

$$\boldsymbol{s}(N_1,N_2)=E_{\hat{\boldsymbol{\sigma}}\sim\mathcal{C}(N_1,N_2)}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{N}\left(\sigma_{i,1}+\sigma_{i,2}\right)\ell\circ f(Z_i)\right]. \tag{59}$$

It is not hard to verify that

$$\begin{aligned}R_{l+u}^{Td}(\mathcal{L}\circ\mathcal{F}\circ\mathbf{Z})&=E_{N_1,N_2}E_{\hat{\boldsymbol{\sigma}}\sim\mathcal{C}(N_1,N_2)}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{N}\left(\sigma_{i,1}+\sigma_{i,2}\right)\ell\circ f(Z_i)\right].\\&=E_{N_1,N_2}s(N_1,N_2)\end{aligned} \tag{60}$$

According to the definition of $\mathbf{H}_k$ and $\mathbf{T}_k$ in Support-Theorem 2, for any $N_1,N_2\in\mathbf{I}_1^{l+u}$, we have

$$\begin{aligned}&E_{\mathbf{Z},\mathbf{Z}'}\sup_{f\in\mathcal{F}}[\mathbf{T}_{N_1}\left\{\mathcal{L}\circ f(\mathbf{Z})\right\}-T_{N_2}\left\{\mathcal{L}\circ f(\mathbf{Z}')\right\}+H_{N_2}\left\{\mathcal{L}\circ f(\mathbf{Z}')\right\}-H_{N_1}\left\{\mathcal{L}\circ f(\mathbf{Z})\right\}]\\=&E_{\mathbf{Z},\mathbf{Z}'}\sup_{f\in\mathcal{F}}[\frac{1}{u}\sum_{i=N_1+1}^{l+u}\ell\circ f(Z_i)-\frac{1}{u}\sum_{i=N_2+1}^{l+u}\ell\circ f(Z_i')+\frac{1}{l}\sum_{i=1}^{N_2}\ell\circ f(Z_i')-\frac{1}{l}\sum_{i=1}^{N_1}\ell\circ f(Z_i)]\end{aligned}. \tag{61}$$

By observing the values of N_1, N_2 and the distribution of $\mathbf{Z}$ and $\mathbf{Z}'$ in Eq.(61), we take the expectation with respect to $\mathbf{Z}$ and $\mathbf{Z}'$. For the convenience of presentation, we introduce a new random variable vector $\boldsymbol{a}=(\boldsymbol{a}_1,\cdots,\boldsymbol{a}_{l+u})$, where $\boldsymbol{a}_i=(a_{i,1},a_{i,2})$ is a pair of coefficients, and let $\mathcal{C}_1(N_1,N_2)$ be the distribution of $\boldsymbol{a}$. Note that the first component of $\boldsymbol{a}_i$ corresponds to $\mathbf{Z}$, and its value is $-\frac{1}{l}$ or $\frac{1}{u}$, and $\mathbf{Z}'$ is assigned to the second component of a_i, which take the values of $\frac{1}{l}$ or $-\frac{1}{u}$. Then we can rewrite Eq.(61) as follows

$$\begin{aligned}&E_{\mathbf{Z},\mathbf{Z}'}\sup_{f\in\mathcal{F}}[\mathbf{T}_{N_1}\left\{\mathcal{L}\circ f(\mathbf{Z})\right\}-\mathbf{T}_{N_2}\left\{\mathcal{L}\circ f(\mathbf{Z}')\right\}+\mathbf{H}_{N_2}\left\{\mathcal{L}\circ f(\mathbf{Z}')\right\}-\mathbf{H}_{N_1}\left\{\mathcal{L}\circ f(\mathbf{Z})\right\}]\\&=E_{\boldsymbol{a}\sim\mathcal{C}_1(N_1,N_2)}\sup_{f\in\mathcal{F}}\left[\sum_{i=1}^{l+u}(a_{i,1}+a_{i,2})\ell\circ f(Z_i)\right]\end{aligned}. \tag{62}$$

Let $\boldsymbol{P}(k)$ be the uniform distribution over partitions of $l+u$ elements into two subsets of k and $l+u-k$ elements, respectively. Clearly, $\boldsymbol{P}(k)$ is a uniform distribution over $\begin{pmatrix} l+u \\ k \end{pmatrix}$ elements. Easy to find that the distribution of the random vector $(a_{1,1},a_{2,1},\cdots,a_{l+u,1})$ of the first elements of pairs in $\boldsymbol{a}$ is equivalent to $\boldsymbol{P}(N_1)$. Similarly, $(a_{1,2},a_{2,2},\cdots,a_{l+u,2})$ is equivalent to $\boldsymbol{P}(N_2)$. Therefore, the distribution $\mathcal{C}_1(N_1,N_2)$ of the entire vector $\boldsymbol{a}$ is equivalent to the product distribution of $\boldsymbol{P}(N_1)$ and $\boldsymbol{P}(N_2)$. Recall the constrained distribution $\mathcal{C}(N_1,N_2)$ of $\hat{\boldsymbol{\sigma}}_i$. We show that the distributions $\mathcal{C}_1(N_1,N_2)$ and $\mathcal{C}(N_1,N_2)$ are identical. For any $N_1,N_2\in\mathbf{I}_1^{l+u}$, let the probability of drawing a specific realization of $\hat{\boldsymbol{\sigma}}$ (under the constrains $n_1+n_2=N_1$ and $n_3+n_2=N_2$)

$$\begin{aligned}\mathcal{Q}(N_1,N_2)&=\left(\frac{l^2}{(l+u)^2}\right)^{n_2}\left(\frac{lu}{(l+u)^2}\right)^{N_1-n_2}\left(\frac{lu}{(l+u)^2}\right)^{N_2-n_2}\left(\frac{u^2}{(l+u)^2}\right)^{l+u-N_1-N_2+n_2}\\&=\frac{l^{N_1+N_2}u^{2(l+u)-N_1-N_2}}{(l+u)^{2(l+u)}}\end{aligned}. \tag{63}$$

Since $\mathcal{Q}(N_1, N_2)$ is independent of n_i, the distribution $\mathcal{C}(N_1, N_2)$ is uniform over all possible Rademacher assignments satisfying the constraints $n_1 + n_2 = N_1$ and $n_3 + n_2 = N_2$. It is not difficult to see that the support size of $\mathcal{C}(N_1, N_2)$ is the same as the support size of $\mathcal{C}_1(N_1, N_2)$. In addition, the support sets of these distributions are the same; therefore, these distributions are identical. Then we further write Eq.(62) as follows

$$\begin{aligned} & E_{\boldsymbol{a}\sim\mathcal{C}_1(N_1,N_2)} \sup_{f\in\mathcal{F}} \left[\sum_{i=1}^{l+u} (a_{i,1} + a_{i,2})\ell \circ f(Z_i)\right] \\ & = E_{\hat{\boldsymbol{\sigma}}_i\sim\mathcal{C}(N_1,N_2)} \left[\sup_{f\in\mathcal{F}} \sum_{i=1}^{N} (\sigma_{i,1} + \sigma_{i,2})\, \ell \circ f(Z_i)\right]. \\ & = \boldsymbol{s}(N_1, N_2) \end{aligned} \tag{64}$$

From Eq.(62) we can find that $E_{\mathbf{Z}}\{\varphi(\mathbf{Z})\}$ is Eq.(61) with $N_1 = l$ and $N_2 = l$, moreover, it is not difficult to find that $E_{\hat{\boldsymbol{\sigma}}} N_1 = E_{\hat{\boldsymbol{\sigma}}}\{n_1 + n_2\} = l$ and $E_{\hat{\boldsymbol{\sigma}}} N_1 = E_{\hat{\boldsymbol{\sigma}}}\{n_2 + n_3\} = l$, we obtain

$$E_{\mathbf{Z}}\{\varphi(\mathbf{Z})\} = E_{\hat{\boldsymbol{\sigma}}_i\sim\mathcal{C}(l,l)} \left[\sup_{f\in F} \sum_{i=1}^{N} (\hat{\sigma}_{i,1} + \hat{\sigma}_{i,2})\, f(Z_i)\right] = \boldsymbol{s}(E_{\hat{\boldsymbol{\sigma}}} N_1, E_{\hat{\boldsymbol{\sigma}}} N_2). \tag{65}$$

We bound the differences $|\boldsymbol{s}(N_1, N_2) - \boldsymbol{s}(N_1', N_2)|$ and $|\boldsymbol{s}(N_1, N_2) - \boldsymbol{s}(N_1', N_2)|$ for any $1 \le N_1, N_2, N_1', N_2' \le l + u$. Suppose that $N_1 \le N_1'$. Recalling the definition of $\boldsymbol{s}(N_1, N_2)$, we have

$$\boldsymbol{s}(N_1', N_2) = \boldsymbol{s}(N_1, N_2) + E_{\mathbf{Z},\mathbf{Z}'} \left[\sup_{f\in\mathcal{F}} \sum_{i=N_1'+1}^{N_1} \left(\frac{1}{u} + \frac{1}{l}\right) \ell \circ f(Z_i)\right]. \tag{66}$$

Therefore, for any N_1 and N_1',

$$\left|\boldsymbol{s}(N_1, N_2) - \boldsymbol{s}(N_1', N_2)\right| \le B\left|N_1 - N_1'\right| \left(\frac{1}{u} + \frac{1}{l}\right). \tag{67}$$

Similarly we have that for any N_2 and N_2',

$$\left|\boldsymbol{s}(N_1, N_2) - \boldsymbol{s}(N_1, N_2')\right| \le B\left|N_2 - N_2'\right| \left(\frac{1}{u} + \frac{1}{l}\right). \tag{68}$$

Noting that $N_1, N_2 \sim \mathcal{B}\left(l + u, \frac{l}{l+u}\right)$, according to Support-Theorem 3, let $n = l + u$ and $p = \frac{l}{l+u}$. Combining Eq.(67) and Eq.(68), we derive the following inequality

$$\begin{aligned} & P_{N_1,N_2}[|\boldsymbol{s}(N_1, N_2) - \boldsymbol{s}(E_{\hat{\boldsymbol{\sigma}}} N_1, E_{\hat{\boldsymbol{\sigma}}} N_2)| \ge \varepsilon] \\ & \le P_{N_1,N_2}[|\boldsymbol{s}(N_1, N_2) - \boldsymbol{s}(N_1, E_{\hat{\boldsymbol{\sigma}}} N_2)| + |\boldsymbol{s}(N_1, E_{\hat{\boldsymbol{\sigma}}} N_2) - \boldsymbol{s}(E_{\hat{\boldsymbol{\sigma}}} N_1, E_{\hat{\boldsymbol{\sigma}}} N_2)| \ge \varepsilon] \\ & \le P_{N_1,N_2}[|\boldsymbol{s}(N_1, N_2) - \boldsymbol{s}(N_1, E_{\hat{\boldsymbol{\sigma}}} N_2)| \ge \frac{\varepsilon}{2}] + P_{N_1,N_2}[|\boldsymbol{s}(N_1, E_{\hat{\boldsymbol{\sigma}}} N_2) - \boldsymbol{s}(E_{\hat{\boldsymbol{\sigma}}} N_1, E_{\hat{\boldsymbol{\sigma}}} N_2)| \ge \frac{\varepsilon}{2}] \\ & \le P_{N_2}[BQ_1\,|N_2 - E_{\hat{\boldsymbol{\sigma}}} N_2| \ge \frac{\varepsilon}{2}] + P_{N_1}[BQ_1\,|N_1 - E_{\hat{\boldsymbol{\sigma}}} N_1| \ge \frac{\varepsilon}{2}] \\ & \le 4\exp\left(\frac{3\varepsilon^2}{32(l+u)\frac{l}{l+u} \cdot B^2\left(\frac{1}{l} + \frac{1}{u}\right)^2}\right) = 4\exp\left(\frac{3\varepsilon^2}{32lB^2\left(\frac{1}{l} + \frac{1}{u}\right)^2}\right) \end{aligned} \tag{69}$$

According to the fact provided in (Devroye et al., 1996), if a non-negative random variable $\mathbf{Z}$ satisfies $P\{\mathbf{Z} > \varepsilon\} \leq c \cdot \exp\left(-k\varepsilon^2\right)$ for some $c \geq 1$ and $k > 0$, then $E\mathbf{Z} \leq \sqrt{\ln(ce)/k}$. Applying this fact with $c = 4$ and $k = \frac{3}{64lQ_1{}^2}$ to Eq.(69), we have

$$\begin{aligned} |E_{N_1,N_2}\{\boldsymbol{s}(N_1,N_2)\} - \boldsymbol{s}(E_{\hat{\boldsymbol{\sigma}}}N_1, E_{\hat{\boldsymbol{\sigma}}}N_2)| &\leq E_{N_1,N_2}|\boldsymbol{s}(N_1,N_2) - \boldsymbol{s}(E_{\hat{\boldsymbol{\sigma}}}N_1, E_{\hat{\boldsymbol{\sigma}}}N_2)| \\ &\leq B\left(\frac{1}{l}+\frac{1}{u}\right)\sqrt{\frac{32l\ln(4e)}{3}}. \end{aligned} \tag{70}$$

Since the entire development is symmetric in l and u, therefore, we also obtain the same result but with u instead of l.

Combining Eq.(60), Eq.(65) and Eq.(70), let $c_0 = \sqrt{\frac{32\ln(4e)}{3}}$, we can obtain

$$E_{\mathbf{Z}}\{\varphi(\mathbf{Z})\} \leq R_{l+u}^{Td}(\mathcal{L}\circ\mathcal{F}\circ\mathbf{Z}) + c_0 B\left(\frac{1}{l}+\frac{1}{u}\right)\sqrt{\min(l,u)}. \tag{71}$$

For a loss function ℓ with Lipschitz constant L_ℓ. Thus, we could apply Support-Theorem 4 and take the expectation of the empirical Rademacher complexity. It holds

$$R_{l+u}^{Td}(\mathcal{L}\circ\mathcal{F}\circ\mathbf{Z}) \leq \sqrt{2}L_\ell(c+1)\Re_{l+u}^{Td}(\mathcal{F}\circ\mathbf{Z}). \tag{72}$$

In addition, according to the constraints $(1-p_i)\boldsymbol{\Omega}\odot\hat{\boldsymbol{y}}_i = \boldsymbol{\Omega}\odot\boldsymbol{y}_i, 0 < p_i < 1, i = l+1,\cdots,n.$ of the LDCL framework, $\hat{e}rr_l(f) = 0$. Substituting Eq.(72) into Eq.(71) and combining with Eq.(57), we can get Eq.(14). Theorem 4 is proved. ■

Appendix C. Proof of Theorem 5

In order to prove Theorem 5, we need to introduce two basic inequalities. Based on Support-Theorem 5 and Support-Theorem 4, we provide the detailed proof of Theorem 5 as follows.
Proof The proof is similar to the proof of Theorem 3.1 in (Jennings and Wooldridge, 2012). For any sample $\mathbf{Z} = (Z_1,\cdots,Z_l,\cdots,Z_{l+u})$, $Z_i = (x_i, y_i)$ and a loss function ℓ with Lipschitz constant L_ℓ and bounded by a constant B, let $\mathcal{L}\circ\mathcal{H}$ correspond the family of loss functions associated to function space $\mathcal{H}$. For any $h \in \mathcal{H}$, we denote $\hat{E}_{\mathbf{Z}}[\mathcal{L}\circ h]$ the empirical average of $\mathcal{L}\circ h$ over $\mathbf{Z}$: $\hat{E}_S[\mathcal{L}\circ h] = \frac{1}{l+u}\sum_{i=1}^{m}\ell\circ h(Z_i)$. Now we defined the function Φ as follows,

$$\Phi(\mathbf{Z}) = \sup_{h\in\mathcal{H}} E[\mathcal{L}\circ h] - \hat{E}_{\mathbf{Z}}[\mathcal{L}\circ h].$$

Let $\mathbf{Z}$ and $\mathbf{Z}'$ be two samples differing by exactly one instance, and say Z_m in $\mathbf{Z}$ and Z'_m in $\mathbf{Z}'$. Then, since the difference of suprema does not exceed the supremum of the difference, we have

$$\Phi(\mathbf{Z}') - \Phi(\mathbf{Z}) \leq \sup_{h\in\mathcal{H}} \hat{E}_{\mathbf{Z}}[\mathcal{L}\circ h] - \hat{E}_{\mathbf{Z}'}[\mathcal{L}\circ h] = \sup_{h\in\mathcal{H}} \frac{\ell\circ(Z_m) - \ell\circ(Z'_m)}{l+u} \leq \frac{B}{l+u}. \tag{73}$$

Similarly, we can obtain $\Phi(\mathbf{Z}) - \Phi(\mathbf{Z}') \leq B/(l+u)$, thus $|\Phi(\mathbf{Z}) - \Phi(\mathbf{Z}')| \leq B/(l+u)$. Then, by McDiarmids inequality, for any $\delta > 0$, with probability at least $1-\delta/2$, the following

holds: $\Phi(\mathbf{Z}) \leq E_{\mathbf{Z}}[\Phi(\mathbf{Z})] + B\sqrt{\frac{\log\frac{2}{\delta}}{2(l+u)}}$. Now we will proceed bound $E_{\mathbf{Z}}[\Phi(\mathbf{Z})]$ as follows

$$
\begin{aligned}
&E_{\mathbf{Z}}[\Phi(\mathbf{Z})] = E_{\mathbf{Z}}[\sup_{h\in\mathcal{H}} E[\mathcal{L}\circ h] - \hat{E}_{\mathbf{Z}}[\mathcal{L}\circ h]] = E_{\mathbf{Z}}[\sup_{h\in\mathcal{H}} E_{\mathbf{Z}'}[E_{\mathbf{Z}'}[\mathcal{L}\circ h] - \hat{E}_{\mathbf{Z}}[\mathcal{L}\circ h]]] \\
&\leq E_{\mathbf{Z},\mathbf{Z}'}[\sup_{h\in\mathcal{H}} \hat{E}_{\mathbf{Z}'}[\mathcal{L}\circ h] - \hat{E}_{\mathbf{Z}}[\mathcal{L}\circ h]] = E_{\mathbf{Z},\mathbf{Z}'}\left[\sup_{h\in\mathcal{H}} \frac{1}{l+u}\sum_{i=1}^{l+u}(\ell\circ h(Z_i') - \ell\circ h(Z_i))\right] \\
&\leq E_{\mathbf{Z},\mathbf{Z}'}[\sup_{h\in\mathcal{H}} \frac{1}{l+u}[\sum_{i=1}^{l}\ell\circ h(Z_i') + \sum_{i=l+1}^{l+u}\big(\ell\circ h(Z_i') + \ell\circ f(Z_i')\big) \\
&\quad - \sum_{i=1}^{l}\ell\circ h(Z_i) - \sum_{i=l+1}^{l+u}(\ell\circ h(Z_i) + \ell\circ f(Z_i))]] \\
&= E_{\boldsymbol{\sigma},\mathbf{Z}'}\left[\sup_{h\in\mathcal{H}} \frac{1}{l+u}\sum_{i=1}^{l+u}\sigma_i\ell\circ h(Z_i')\right] + E_{\boldsymbol{\sigma},\mathbf{Z}}\left[\sup_{h\in\mathcal{H}} \frac{1}{l+u}\sum_{i=1}^{l+u} -\sigma_i\ell\circ h(Z_i)\right] \\
&+ E_{\boldsymbol{\sigma},\mathbf{Z}'}\left[\sup_{h\in\mathcal{H}} \frac{1}{l+u}\sum_{i=l+1}^{l+u}\sigma_i\ell\circ f(Z_i')\right] + E_{\boldsymbol{\sigma},\mathbf{Z}}\left[\sup_{h\in\mathcal{H}} \frac{1}{l+u}\sum_{i=l+1}^{l+u} -\sigma_i\ell\circ f(Z_i')\right] \\
&= 2E_{\boldsymbol{\sigma},\mathbf{Z}}\left[\sup_{h\in\mathcal{H}} \frac{1}{l+u}\sum_{i=1}^{l+u}\sigma_i\ell\circ h(Z_i)\right] + 2\frac{u}{l+u}E_{\boldsymbol{\sigma},\mathbf{Z}}\left[\sup_{h\in\mathcal{H}} \frac{1}{l+u}\sum_{i=1}^{l+u}\sigma_i\ell\circ f(Z_i)\right] \\
&= 2\Re_{l+u}(\mathcal{L}\circ\mathcal{H}) + 2\frac{u}{l+u}\Re_u(\mathcal{L}\circ\mathcal{F}).
\end{aligned}
\tag{74}
$$

Thus, we have

$$
R(h) \leq \hat{R}(h) + 2\Re_{l+u}(\mathcal{L}\circ\mathcal{H}) + 2\frac{u}{l+u}\Re_u(\mathcal{L}\circ\mathcal{F}) + B\sqrt{\frac{\log 1/\delta}{2(l+u)}}. \tag{75}
$$

For a loss function ℓ with Lipschitz constant L_ℓ. Thus, we could apply Support-Theorem 4 and take the expectation of the empirical Rademacher complexity. It holds

$$
\Re_{l+u}(\mathcal{L}\circ\mathcal{H}) \leq \sqrt{2}L_\ell(c+1)\Re_{l+u}(\mathcal{H}), \quad \Re_u(\mathcal{L}\circ\mathcal{F}) \leq \sqrt{2}L_\ell(c+1)\Re_u(\mathcal{F}) \ . \tag{76}
$$

In addition, since h is trained from the recovered label distribution matrix, $\hat{R}(h) \leq \hat{err}_{l+u}(h) + \hat{err}_u(f)$. Substituting Eq.(76) into Eq.(75), we can get Eq.(15), Theorem 5 is proved. ■

Appendix D. Supplementary Experiment

In this part, we will add some supplementary experiments to demonstrate the LDCL model framework.

D.1 Empirical Comparison and Discussion of Classification Models

In this section, we compare and discuss the two classification models under the LDCL framework, mainly analyzing the performance of the classification model combined with

Table 10: Chebyshev ("↓")(mean±std) results on test set. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	KL+softmax	KL+LN	L_2+softmax	L_2+LN
Yeast-alpha	**.0141±.0000**	.0144±.0001	.0148±.0001	.0143±.0002
Yeast-cdc	**.0162±.0000**	.0164±.0001	.0168±.0001	.0165±.0003
Yeast-elu	**.0160±.0001**	.0166±.0001	.0165±.0002	.0167±.0001
Yeast-diau	**.0408±.0001**	.0413±.0013	.0414±.0002	.0412±.0001
Yeast-heat	**.0426±.0001**	.0474±.0006	.0463±.0003	.0449±.0005
win/tie/loss	5/0/0	0/0/5	0/0/5	0/0/5

softmax transformation and linear normalization (LN). The experiment results are shown in Table 10. From the experiment results, it can be seen that the maximum entropy model ("KL+softmax") has the best performance. Compared with "KL + softmax", the performance of "KL + LN" is reduced. Similarly, the performance of "L_2+softmax" is degenerated compared with "L_2 + LN". The above phenomena indicate that the maximum entropy model with KL divergence is more targeted for LDL.

D.2 Experiments with more models under the LDCL framework

In this section, we compare and analyze more specific models under the LDCL model framework. Inspired by the investigation in (Lv et al., 2019) and (Xu et al., 2021b), we know that compared to the pure graph-based or feature-manifold-based methods, the method that combines feature structure and label correlation can achieve superior performance in recovering the label distribution matrix. Therefore, for illustration and abbreviation, we mainly analyze and compare several representative methods in 18 concrete models under the LDCL framework experimentally. We choose the pure graph based methods as the baseline and combined methods as the advanced approaches. By instantiating them into our framework, we conduct experiments on the 'Yest-heat' dataset. The experiment results are shown in Table 11. According to the experiment results, it can be concluded that our framework is effective in modifying traditional LDL methods. Besides, the recovery performance of the three manifold enhancement methods for the expanded label distribution matrix is better than that of the two graph methods, which consists with the conclusion in (Lv et al., 2019). Certainly, it is still a valuable research topic to perform other methods and we leave them as the furture work.

D.3 Experiment under more metrics

In this section we added experiment results under three metrics Clark, Canberra and Cosine to evaluate the LD framework from several different perspectives. The experiment results are shown in Table 12-Table 17. From the experimental results, it can be found that the conclusions drawn by Clark, Canberra and Cosine are basically consistent with those drawn by Chebyshev and Intersection. Although there are minor differences in individual datasets,

Table 11: Experiment results(mean±std) in setting 1 on un-relabeled data for 'Yeast-heat' dataset.

Methods	Chebyshev	Clark	Canberra	Cosine	Intersection	win/tie/loss
Graph+$L2$+$\|\boldsymbol{p}\|_2^2$	.0347±.0017	.1124±.0049	.2069±.0096	.9936±.0006	.9653±.0017	4/0/0
Graph+KL+$(-\boldsymbol{p}^\top \log \boldsymbol{p})$	.0345±.0011	.1117±.0032	.2056±.0062	.9937±.0003	.9655±.0011	
Manifold+$L2$+$\|\boldsymbol{p}\|_2^2$	.0251±.0022	.0844±.0064	.1523±.0123	.9963±.0005	.9749±.0022	
Manifold+KL+$\|\boldsymbol{p}\|_2^2$	.0258±.0020	.0885±.0075	.1585±.0128	.9962±.0004	.9742±.0020	
Manifold+KL+$(-\boldsymbol{p}^\top \log \boldsymbol{p})$	.0277±.0026	.0924±.0049	.1678±.0137	.9955±.0006	.9723±.0026	
BFGS-LDL	.0447±.0007	.1923±.0029	.3844±.0062	.9866±.0004	.9368±.0010	0/0/4
PT-SVM	.0476±.0026	.2069±.0130	.4152±.0173	.9847±.0016	.9318±.0044	0/0/4
PT-Bayes	.4302±.0570	.9712±.0311	.9298±.3080	.8481±.0446	.5057±.0500	0/0/4
IIS-LDL	.0528±.0004	.2249±.0014	.4570±.0031	.9812±.0003	.9238±.0005	0/0/4

Table 12: Clark ("↓")(mean±std) results in setting 1 on un-relabeled data. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
Yeast-alpha	.0358±.0002	**.0356±.0005**	.2237±.0019	.2545±.0124	.6137±.1329	.3026±.0014
Yeast-cdc	.0469±.0007	**.0454±.0003**	.2336±.0014	.2488±.0084	.6286±.0837	.2951±.0011
Yeast-elu	.0557±.0020	**.0548±.0004**	.2111±.0019	.2485±.0252	.6387±.0864	.2762±.0014
Yeast-diau	.1310±.0053	**.1210±.0046**	.2116±.0031	.2460±.0112	.5814±.0599	.2426±.0013
Yeast-heat	.1117±.0032	**.0844±.0064**	.1923±.0029	.2069±.0130	.9712±.0311	.2249±.0014
Yeast-spo	.1478±.0012	**.1214±.0029**	.2627±.0031	.2759±.0066	.7449±.0531	.2799±.0015
Yeast-cold	**.0893±.0003**	.0942±.0019	.1462±.0012	.1569±.0070	.6146±.0928	.1643±.0009
Yeast-dtt	.0599±.0021	**.0519±.0037**	.1532±.0047	.1578±.0144	.5052±.0344	.1773±.0040
Yeast-spo5	**.1468±.0005**	.1524±.0014	.2387±.0079	.2445±.0134	.4741±.0369	.2465±.0075
Yeast-spoem	.1317±.0011	**.1314±.0015**	.2464±.0080	.2702±.0344	.3516±.0198	.2484±.0076
emotion6	.8074±.0019	**.7068±.0014**	1.774±.1972	1.728±.0314	1.680±.0031	1.659±.0034
Natural scene	**.4754±.0012**	.9823±.0012	2.707±.0309	2.556±.0260	2.485±.0035	2.474±.0043
MovieDataSet	.3346±.0017	**.3012±.0005**	.6509±.0104	.8735±.0770	.7516±.0007	.6239±.0049
win/tie/loss	3/0/10	10/0/3	0/0/13	0/0/13	0/0/13	0/0/13

the performance of the proposed method is better than that of the baseline methods, no matter which metric is employed. It further illustrates the effectiveness of the proposed LDCL framework.

Table 13: Canberra ("↓")(mean±std) results in setting 1 on un-relabeled data. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
Yeast-alpha	.0702±.0006	**.0694±.0003**	.7300±.0047	.8420±.0448	.8567±.1354	1.011±.0050
Yeast-cdc	.0875±.0004	**.0866±.0006**	.7050±.0038	.7513±.0254	.8853±.3228	.9048±.0032
Yeast-elu	.2161±.0010	**.1843±.0004**	.6208±.0060	.7368±.0773	.7909±.3520	.8266±.0049
Yeast-diau	.2315±.0097	**.2133±.0081**	.4546±.0073	.5292±.0305	.8895±.1619	.5299±.0027
Yeast-heat	.2056±.0062	**.1523±.0123**	.3844±.0062	.4152±.0173	.9298±.3080	.4570±.0031
Yeast-spo	.2886±.0012	**.2806±.0039**	.5470±.0070	.5698±.0144	.8243±.1331	.5757±.0033
Yeast-cold	**.1463±.0013**	.1576±.0024	.2521±.0023	.2704±.0114	.9107±.1819	.2849±.0017
Yeast-dtt	.1033±.0037	**.0890±.0062**	.2563±.0074	.2660±.0222	.8905±.0641	.3012±.0061
Yeast-spo5	**.2162±.0005**	.2168±.0004	.3652±.0117	.3741±.0207	.7305±.0608	.3772±.0109
Yeast-spoem	.1817±.0015	**.1811±.0021**	.3426±.0109	.3466±.0487	.6746±.0270	.3451±.0104
emotion6	.1.145±.0019	**1.039±.0004**	4.112±.6402	3.985±.1083	3.828±.0084	3.735±.0120
Natural scene	**.5956±.0095**	1.203±.0041	7.836±.1176	7.220±.1329	7.001±.0131	6.826±.0207
MovieDataSet	.0845±.0016	**.0772±.0005**	1.265±.0231	1.690±.1833	1.434±.0014	1.201±.0085
win/tie/loss	3/0/10	10/0/3	0/0/13	0/0/13	0/0/13	0/0/13

Table 14: Cosine ("↑")(mean±std) results for setting 1 on un-relabeled data. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
Yeast-alpha	.9994±.0004	**.9998±.0000**	.9939±.0001	.9924±.0007	.8387±.0319	.9882±.0003
Yeast-cdc	.9989±.0001	**.9994±.0003**	.9923±.0001	.9913±.0005	.8484±.0244	.9868±.0001
Yeast-elu	.9989±.0001	**.9992±.0001**	.9933±.0001	.9909±.0017	.7626±.0202	.9875±.0002
Yeast-diau	.9939±.0005	**.9949±.0004**	.9864±.0005	.9823±.0018	.8309±.0262	.9819±.0002
Yeast-heat	.9937±.0004	**.9963±.0006**	.9866±.0004	.9847±.0016	.8481±.0446	.9812±.0003
Yeast-spo	.9824±.0009	**.9950±.0019**	.9742±.0007	.9710±.0015	.7487±.0172	.9711±.0004
Yeast-cold	**.9927±.0005**	.9925±.0005	.9875±.0002	.9856±.0012	.7985±.0292	.9838±.0002
Yeast-dtt	.9972±.0001	**.9973±.0001**	.9881±.0006	.9870±.0017	.8907±.0104	.9833±.0005
Yeast-spo5	**.9789±.0005**	.9786±.0017	.9604±.0020	.9588±.0034	.8817±.0142	.9568±.0020
Yeast-spoem	.9787±.0004	**.9789±.0005**	.9360±.0031	.9222±.0190	.9004±.0088	.9353±.0030
emotion6	.9866±.0004	**.9869±.0003**	.6133±.0954	.6169±.0345	.6564±.0008	.6951±.0033
Natural scene	**.9847±.0006**	.9841±.0012	.3896±.0259	.4885±.0471	.5750±.0014	.6616±.0050
MovieDataSet	.9719±.0005	**.9753±.0013**	.8894±.0040	.7618±.0513	.8591±.0002	.8933±.0016
win/tie/loss	3/0/10	10/0/3	0/0/13	0/0/13	0/0/13	0/0/13

Table 15: Clark ("↓")(mean±std) results for setting 2 on test set. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
Yeast-alpha	.2182±.0000	**.2178±.0001**	.2298±.0022	.2589±.0122	2.825±.1360	.3158±.0006
Yeast-cdc	.2094±.0000	**.2090±.0001**	.2213±.0029	.2283±.0083	2.520±.0892	.2871±.0007
Yeast-elu	.2052±.0000	**.2047±.0001**	.2108±.0021	.2499±.0257	2.329±.0757	.2755±.0014
Yeast-diau	**.2143±.0004**	.2159±.0008	.2190±.0039	.2451±.0119	1.603±.0581	.2404±.0007
Yeast-heat	.1873±.0002	**.1868±.0002**	.1967±.0022	.2089±.0140	1.517±.1247	.2210±.0011
Yeast-spo	.2454±.0002	**.2415±.0006**	.2559±.0040	.2650±.0048	1.531±.0567	.2634±.0004
Yeast-cold	.1511±.0004	**.1502±.0007**	.1844±.0041	.2099±.0243	.5174±.0551	.2077±.0037
Yeast-dtt	.0998±.0002	**.0981±.0003**	.1459±.0044	.1561±.0169	.4799±.0346	.1677±.0039
Yeast-spo5	.1807±.0032	**.1716±.0010**	.2235±.0079	.2312±.0153	.4613±.0348	.2366±.0072
Yeast-spoem	.1263±.0008	**.1258±.0009**	.2547±.0076	.2704±.0261	.3533±.0206	.2597±.0076
emotion6	1.654±.0006	**1.507±.0011**	1.766±.2071	1.713±.0106	1.670±.0000	1.643±.0020
Natural scene	**2.464±.0007**	2.465±.0006	2.701±.0248	.2.555±.0231	2.479±.0000	2.478±.0040
MovieDataSet	**.6243±.0002**	.6314±.0004	.6538±.0131	.8803±.0760	.7588±.0000	.6338±.0038
win/tie/loss	3/0/10	10/0/3	0/0/13	0/0/13	0/0/13	0/0/13

Table 16: Canberra ("↓")(mean±std) results for setting 2 on test set. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
Yeast-alpha	.7086±.0001	**.7072±.0000**	.7488±.0076	.8559±.0440	6.700±.0992	1.061±.0021
Yeast-cdc	.6279±.0001	**.6267±.0001**	.6729±.0088	.7239±.0264	6.872±.3539	.8846±.0046
Yeast-elu	.6133±.0001	**.6116±.0001**	.6279±.0060	.7462±.0746	6.979±.3046	.8233±.0039
Yeast-diau	.4517±.0062	**.4346±.0014**	.4519±.0090	.5263±.0305	3.861±.1530	.5259±.0019
Yeast-heat	.3749±.0005	**.3739±.0003**	.3944±.0051	.4191±.0276	3.413±.3246	.4479±.0021
Yeast-spo	.5059±.0004	**.4985±.0012**	.5286±.0090	.5454±.0111	3.451±.1505	.5651±.0034
Yeast-cold	.2573±.0003	**.2559±.0013**	.3106±.0064	.3554±.0041	.9009±.0989	.3532±.0055
Yeast-dtt	.1734±.0004	**.1704±.0009**	.2459±.0068	.2661±.0260	.8437±.0678	.2848±.0059
Yeast-spo5	.2750±.0047	**.2630±.0015**	.3422±.0117	.3535±.0239	.7158±.0575	.3621±.0109
Yeast-spoem	.1761±.0012	**.1753±.0013**	.3547±.0104	.3770±.0366	.4780±.0282	.3611±.0103
emotion6	3.765±.0002	**3.214±.0001**	4.105±.6673	3.940±.1059	3.830±.0000	3.702±.0130
Natural scene	**6.342±.0023**	6.889±.0016	7.808±.1003	7.209±.1195	6.973±.0000	6.835±.0243
MovieDataSet	**1.130±.0018**	1.169±.0017	1.271±.0296	1.705±.1841	1.450±.0000	1.219±.0073
win/tie/loss	2/0/11	11/0/2	0/0/13	0/0/13	0/0/13	0/0/13

Table 17: Cosine ("↑")(mean±std) results for setting 2 on test set. The best results on each row are bolded, together with pairwise single-tailed t-test at 95% confidence level.

Dataset	Graph	Manifold	BFGS-LDL	PT-SVM	PT-Bayes	IIS-LDL
Yeast-alpha	.9941±.0000	**.9942±.0000**	.9935±.0001	.9920±.0007	.6205±.0130	.9871±.0006
Yeast-cdc	.9933±.0000	**.9934±.0000**	.9927±.0001	.9917±.0005	.5435±.0243	.9872±.0001
Yeast-elu	.9937±.0001	**.9938±.0000**	.9933±.0001	.9908±.0017	.5763±.0201	.9877±.0001
Yeast-diau	**.9864±.0005**	.9862±.0001	.9858±.0006	.9825±.0017	.6331±.0263	.9825±.0001
Yeast-heat	.9872±.0000	**.9873±.0001**	.9862±.0003	.9845±.0017	.6336±.0463	.9821±.0001
Yeast-spo	.9765±.0004	**.9772±.0001**	.9750±.0009	.9730±.0011	.6259±.0226	.9724±.0002
Yeast-cold	.9865±.0001	**.9867±.0001**	.9824±.0006	.9767±.0051	.8878±.0154	.9767±.0006
Yeast-dtt	.9939±.0000	**.9940±.0000**	.9891±.0005	.9872±.0021	.8976±.0119	.9852±.0005
Yeast-spo5	.9752±.0007	**.9769±.0002**	.9639±.0019	.9623±.0032	.8733±.0133	.9591±.0019
Yeast-spoem	.9791±.0003	**.9793±.0002**	.9321±.0031	.9227±.0140	.8990±.0093	.9314±.0031
emotion6	.6846±.0005	**.6986±.0004**	.6230±.1084	.6200±.0340	.6582±.0000	.7010±.0044
Natural scene	**.6548±.0010**	.6332±.0003	.3982±.0259	.4838±.0474	.5750±.0000	.6529±.0056
MovieDataSet	**.8926±.0004**	.8891±.0004	.8889±.0042	.7589±.0516	.8575±.0000	.8923±.0016
win/tie/loss	3/0/10	10/0/3	0/0/13	0/0/13	0/0/13	0/0/13

References

Rana Ali Amjad. *Applications of Information Theory and Factor Graphs for Machine Learning*. PhD thesis, Technical University of Munich, Germany, 2019.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2002.

Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20(4):1956–1982, 2010.

Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulò, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9230–9239. Computer Vision Foundation / IEEE, 2020.

Yi-Liang Chen and Wei-Hou Cheng. A note on joint distributions of some random vectors defined on permutations. *Ars Comb.*, 54, 1999.

Robert Craigen, Jennifer Seberry, and Xian-Mo Zhang. Product of four hadamard matrices. *J. Comb. Theory, Ser. A*, 59(2):318–320, 1992.

Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Stochastic Modelling and Applied Probability*. Springer, 1996.

Ran El-Yaniv and Dmitry Pechyony. Stable transductive learning. In Gábor Lugosi and Hans Ulrich Simon, editors, *Learning Theory, 19th Annual Conference on Learning Theory, COLT 2006*, volume 4005 of *Lecture Notes in Computer Science*, pages 35–49. Springer, 2006.

Ran El-Yaniv and Dmitry Pechyony. Transductive rademacher complexity and its applications. *J. Artif. Intell. Res.*, 35:193–234, 2009.

Michael Fairbank, Spyridon Samothrakis, and Luca Citi. Deep learning in target space. *J. Mach. Learn. Res.*, 23:8:1–8:46, 2022.

Bin Bin Gao, Chao Xing, Chen Wei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, PP(99):1–1, 2016.

Xin Geng. Label distribution learning. *IEEE Trans. Knowl. Data Eng.*, 28(7):1734–1748, 2016.

Xin Geng, Kate Smith-Miles, and Zhi-Hua Zhou. Facial age estimation by learning from label distributions. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.

Xin Geng, Chao Yin, and Zhi-Hua Zhou. Facial age estimation by learning from label distributions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(10):2401–2412, 2013.

Peng Hou, Xin Geng, and Min-Ling Zhang. Multi-label manifold learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1680–1686. AAAI Press, 2016.

N. R. Jennings and M. J. Wooldridge. *Foundations of Machine Learning*. Foundations of machine learning, 2012.

Xiuyi Jia, Tingting Ren, Lei Chen, Jun Wang, Jihua Zhu, and Xianzhong Long. Weakly supervised label distribution learning based on transductive matrix completion with sample correlations. *Pattern Recognition Letters*, 125:453–462, 2019.

Yu-Feng Li, Lan-Zhe Guo, and Zhi-Hua Zhou. Towards safe weakly supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1):334–346, 2021.

Yu-Kun Li, Min-Ling Zhang, and Xin Geng. Leveraging implicit relative labeling-importance information for effective multi-label learning. In *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*, pages 251–260. IEEE Computer Society, 2015.

Jiaqi Lv, Ning Xu, RenYi Zheng, and Xin Geng. Weakly supervised multi-label learning via label enhancement. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3101–3107. ijcai.org, 2019.

Andreas Maurer. A vector-contraction inequality for rademacher complexities. In Ronald Ortner, Hans Ulrich Simon, and Sandra Zilles, editors, *Algorithmic Learning Theory - 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings*, volume 9925 of *Lecture Notes in Computer Science*, pages 3–17, 2016.

C. Mcdiarmid. *On the method of bounded differences.* Surveys in Combinatorics, 1989.

Xin Mu, Kai Ming Ting, and Zhi-Hua Zhou. Classification under streaming emerging new classes: A solution using completely random trees. *CoRR*, abs/1605.09131, 2016.

Cheong Hee Park and Hongsuk Shim. Detection of an emerging new class using statistical hypothesis testing and density estimation. *Int. J. Pattern Recognit. Artif. Intell.*, 24(1): 1–14, 2010.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5533–5542. IEEE Computer Society, 2017.

Martin Rudorfer. *Towards robust object detection and pose estimation as a service for manufacturing industries.* PhD thesis, Technical University of Berlin, Germany, 2021.

Richard P. Stanley. Alternating permutations and symmetric functions. *J. Comb. Theory, Ser. A*, 114(3):436–460, 2007.

J. Wang and X. Geng. Label distribution learning by exploiting label distribution manifold. *IEEE Transactions on Neural Networks and Learning Systems*, PP, 2021.

Ke Wang, Ning Xu, Miaogen Ling, and Xin Geng. Fast label enhancement for label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021. doi: 10.1109/TKDE.2021.3092406.

Xiu-Shen Wei, Han-Jia Ye, Xin Mu, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Multi-instance learning with emerging novel class. *IEEE Trans. Knowl. Data Eng.*, 33(5):2109–2120, 2021.

Chao Xing, Xin Geng, and Hui Xue. Logistic boosting regression for label distribution learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4489–4497. IEEE Computer Society, 2016.

Chao Xu, Shilin Gu, Hong Tao, and Chenping Hou. Fragmentary label distribution learning via graph regularized maximum entropy criteria. *Pattern Recognit. Lett.*, 145:147–156, 2021a.

Miao Xu and Zhi-Hua Zhou. Incomplete label distribution learning. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3175–3181. ijcai.org, 2017.

Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2301–2309, 2013.

Ning Xu, Yun-Peng Liu, and Xin Geng. Label enhancement for label distribution learning. *IEEE Trans. Knowl. Data Eng.*, 33(4):1632–1643, 2021b.

Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J. Rosato. A 3d facial expression database for facial behavior research. In *Seventh IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2006), 10-12 April 2006, Southampton, UK*, pages 211–216. IEEE Computer Society, 2006.

Peng Zhao and Zhi-Hua Zhou. Label distribution learning by optimal transport. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 4506–4513. AAAI Press, 2018.

Deyu Zhou, Xuan Zhang, Yin Zhou, Quan Zhao, and Xin Geng. Emotion distribution learning from texts. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 638–647. The Association for Computational Linguistics, 2016.

Ying Zhou, Hui Xue, and Xin Geng. Emotion distribution recognition from facial expressions. In Xiaofang Zhou, Alan F. Smeaton, Qi Tian, Dick C. A. Bulterman, Heng Tao Shen, Ketan Mayer-Patel, and Shuicheng Yan, editors, *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM '15, Brisbane, Australia, October 26 - 30, 2015*, pages 1247–1250. ACM, 2015.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

Yue Zhu, James T. Kwok, and Zhi-Hua Zhou. Multi-label learning with global and local label correlation. *IEEE Trans. Knowl. Data Eng.*, 30(6):1081–1094, 2018a.

Yue Zhu, Kai Ming Ting, and Zhi-Hua Zhou. Multi-label learning with emerging new labels. *IEEE Trans. Knowl. Data Eng.*, 30(10):1901–1914, 2018b.