Variational Label Enhancement

Ning Xu, *Member, IEEE*, Jun Shu, Renyi Zheng, Xin Geng*, *Senior Member, IEEE*, Deyu Meng, *Member, IEEE*, and Min-Ling Zhang, *Senior Member, IEEE*

Abstract—Multi-label learning focuses on the ambiguity at the label side, i.e., one instance is associated with multiple class labels, where the logical labels are always adopted to partition class labels into relevant labels and irrelevant labels rigidly. However, the relevance or irrelevance of each label corresponding to one instance is essentially relative in real-world tasks and the label distribution is more fine-grained than the logical labels by denoting one instance with a certain number of the description degrees of all class labels. As the label distribution is not explicitly available in most training sets, a process named label enhancement emerges to recover the label distributions in training datasets. By inducing the generative model of the label distribution and adopting the variational inference technique, the approximate posterior density of the label distributions should maximize the variational lower bound. Following the above consideration, LEVI is proposed to recover the label distributions from the training examples. In addition, the multi-label predictive model is induced for multi-label learning by leveraging the recovered label distributions along with a specialized objective function. The recovery experiments on fourteen label distribution datasets and the predictive experiments on fourteen multi-label learning datasets validate the advantage of our approach over the state-of-the-art approaches.

Index Terms—Label enhancement, label distribution learning, multi-label learning, label ambiguity.

1 Introduction

Learning with ambiguity is a hot topic in recent machine learning and data mining research. The paradigm of multi-label learning (MLL) naturally emerges, which focuses on the label ambiguity, i.e., one instance is associated with multiple class labels [18], [44], [57]. During the past decade, multi-label learning has been widely adopted to learn from data with rich semantics, such as image [3], [46], text [6], [36], audio [21], [27], video [21], [47], etc. Logical labels are always assigned to the instances in multi-label learning, which partition the labels into relevant labels and irrelevant labels rigidly.

Actually, the relevance or irrelevance of each label corresponding to one instance is essentially relative in real-world tasks. If an instance is denoted with multiple labels, the relative importance among the labels is more likely to be different rather than exactly equal. For example, when "sailboat" and "sand" are relevant to the two images in Fig. 1, "sand" is more significant than "sailboat" in image (a) and the opposite scenario occurs in image (b). On the other hand, the "irrelevance" of irrelevant labels may also be different. For example, "bus" is more irrelevant than "sun" to the two images in Fig. 1 as "sun" often appears with "sand" and "sailboat" on the beach. Therefore, assigning the logical label $l_{\boldsymbol{x}}^{y} \in \{0, 1\}$ to each instance $\boldsymbol{x}$ with class label y ignores the relative importance among the relevant (or irrelevant) labels.

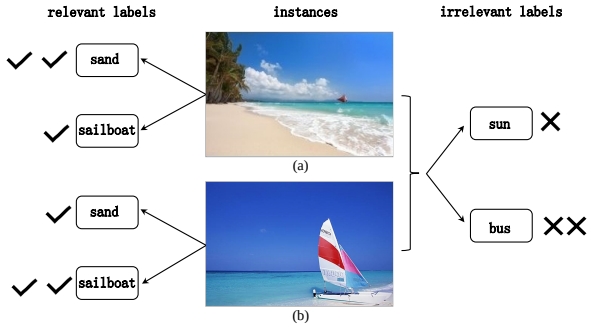


Fig. 1: An example of the relative importance among relevant and irrelevant labels

Therefore, a more natural way to denote the supervised information of $\boldsymbol{x}$ is assigning a real-valued $d_{\boldsymbol{x}}^{y}$ to each label y, which represents the degree to which y describes $\boldsymbol{x}$. Such $d_{\boldsymbol{x}}^{y}$ is called the *description degree* of y to $\boldsymbol{x}$. For a particular instance, the real-valued vector constituted by the description degrees of all the labels is called *label distribution* [14], [50]. Therefore, the label distribution is more fine-grained than logical labels for describing the supervised information in the tasks of learning with label ambiguity.

However, given the difficulty and cost of quantifying the description degrees, the label distributions are not explicitly available in most training sets. They need to be recovered from the training set via the process named label enhancement (LE) [52]. Then, more effective supervised learning can be achieved by leveraging the recovered label distributions rather than learning directly on the original logical labels [19], [55]. In recent years, label enhancement has been suc-

- *Ning Xu, Renyi Zheng, Xin Geng and Min-Ling Zhang are with the School of Computer Science and Engineering, and the Key Lab of Computer Network and Information Integration (Ministry of Education), Southeast University, Nanjing 211189, China.*
  E-mail: {xning, zhengry, xgeng, zhangml}@seu.edu.cn
- *Jun Shu is with the School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an, Shaanxi, China.*
  E-mail: xjtushujun@gmail.com
- *Deyu Meng is with the School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an, Shaanxi, China, and Macao Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macao.*
  E-mail: dymeng@mail.xjtu.edu.cn

*Corresponding author.

cessfully employed for multi-label learning [28], [33], [37], [54], label distribution learning [13], [41], [42], facial emotion recognition [4], cross-modal retrieval [25], etc. Although some label enhancement methods [26], [43], [49], [54], [61] have been proposed, there is no theoretical explanation about the process of recovering label distributions.

In this paper, a theoretical explanation about the essence of label enhancement is proposed. By inducing the generative model of the label distribution and adopting the variational inference technique, the approximate posterior density of the label distributions should maximize the variational lower bound. Following the above consideration, Label Enhancement via Variational Inference (LEVI) is proposed to recover the label distributions from the training examples. Specifically, the approximate posterior density is constructed by employing a multi-layer perceptron or graph convolutional network, which is optimized with the variational lower bound. In addition, the multi-label predictive model would be induced for multi-label learning by leveraging the recovered label distributions along with a specialized objective function. Comprehensive experimental studies validate the performance superiority of the proposed approaches against state-of-the-art comparing approaches as well as the usefulness of the recovered label distributions.

Preliminary results of this paper have been reported in a shorter conference version [51]. While only the variational lower bound is employed to recover label distributions, here we consider exploring the topological information and propose another LE method which constructs an approximate posterior density with the explored topological information by employing a graph convolutional network. Moreover, the details about the specialized objective function to achieve effective multi-label learning are shown, which induces a multi-label predictive model with original logical labels and the recovered label distributions. Besides, more datasets and comparing algorithms are added into the recovery experiment and the predictive experiment, and the ablation experiment on MLL datasets is conducted to show the usefulness of the proposed LE approaches.

The rest of this paper is organized as follows. Firstly, some related work is briefly reviewed and discussed in Section 2. Secondly, technical details of the theoretical explanation and proposed method LEVI for LE and MLL are introduced in Section 3. After that, the results of the label distribution recovery experiments and the MLL prediction experiments are reported in Section 4. Finally, we conclude this paper in Section 5.

2 Related Work

To deal with label ambiguity, multi-label learning is to learn a mapping from the instance space to the power set of the label space. Logical labels are assigned to the instance in multi-label learning, which partition the supervised information into relevance and irrelevance labels rigidly. The simplest approaches are designed to decompose the multi-label classification problem into a series of binary classification problems for each class [1], [56], i.e., each class is independently considered. The correlations between pairs of classes are considered in [10], [11], which focus on the difference between the relevant label and the irrelevant label. Furthermore, the correlations among label subsets or all the class labels are considered in [34], [45]. Some works learn from multi-label data with auxiliary importance of labels, which is explicitly given and accessible to the learning approaches. For example, an ordinal scale is considered to characterize the degrees of labels, and the ordinal grades of labels are assigned to the training examples [2], [10]. A full ordering assigned to the training examples is considered to rank relevant labels [48].

Label distribution is the real-valued vector constituted by the description degree, which represents the degree of each label y describing an instance $\boldsymbol{x}$. Therefore, label distribution is more fine-grained than logical labels for describing the supervised information in the tasks of learning with label ambiguity. The paradigm of label distribution learning (LDL) labels an instance with a label distribution and learns a mapping from an instance to a label distribution straightly. IIS-LDL and BFGS-LDL [14] are the representative LDL approaches, which adopt the maximum entropy model for learning the label distributions. In addition, [15] proposes a SVR-based approach to deal with LDL. Furthermore, [38] extends random forest to learn label distribution. Label distribution learning has been successfully adopted to deal with many real applications, such as age estimation [12], emotion analysis [60], facial landmark detection [40], and multi-label ranking [16].

Label enhancement (LE) is a process to recover the label distributions from the training examples. Graph-Laplacian-based LE method [49] constructs a local similarity matrix to preserve the structure of the feature space and transfers logical labels into label distributions with the local similarity matrix. The label propagation technique is employed in [55] to propagate labeling-importance information and generate the label distributions. Manifold base LE approach [19] adopts the locally linear embedding technique to achieve identified label distributions. Tang [43] proposes a low-rank representation LE method via capturing the global relationships of samples and predicting the implicit label correlation. Zhu [61] adopts the structural information between instances and the privileged information to recover label distributions. A bi-directional loss function [26] is proposed to fully explore the relationship between the feature space and the label distribution space. In recent years, label enhancement has been successfully employed for multi-label learning [28], [33], [37], [54], label distribution learning [13], [41], [42], facial emotion recognition [4], cross-modal retrieval [25], etc.

In the next section, a theoretical explanation about the essence of label enhancement is proposed. By inducing the generative model of the label distribution and adopting the variational inference technique, the posterior density of the label distributions should maximize the variational lower bound. Following the above consideration, LEVI is proposed to recover the label distributions from the logical labels. Different from existing label enhancement approaches, the approximate posterior density is constructed by employing a multi-layer perceptron or a graph convolutional network, which is optimized with the variational lower bound. In addition, the multi-label predictive model is induced for multi-label learning by fitting a predictive model with logical labels and recovered label distributions

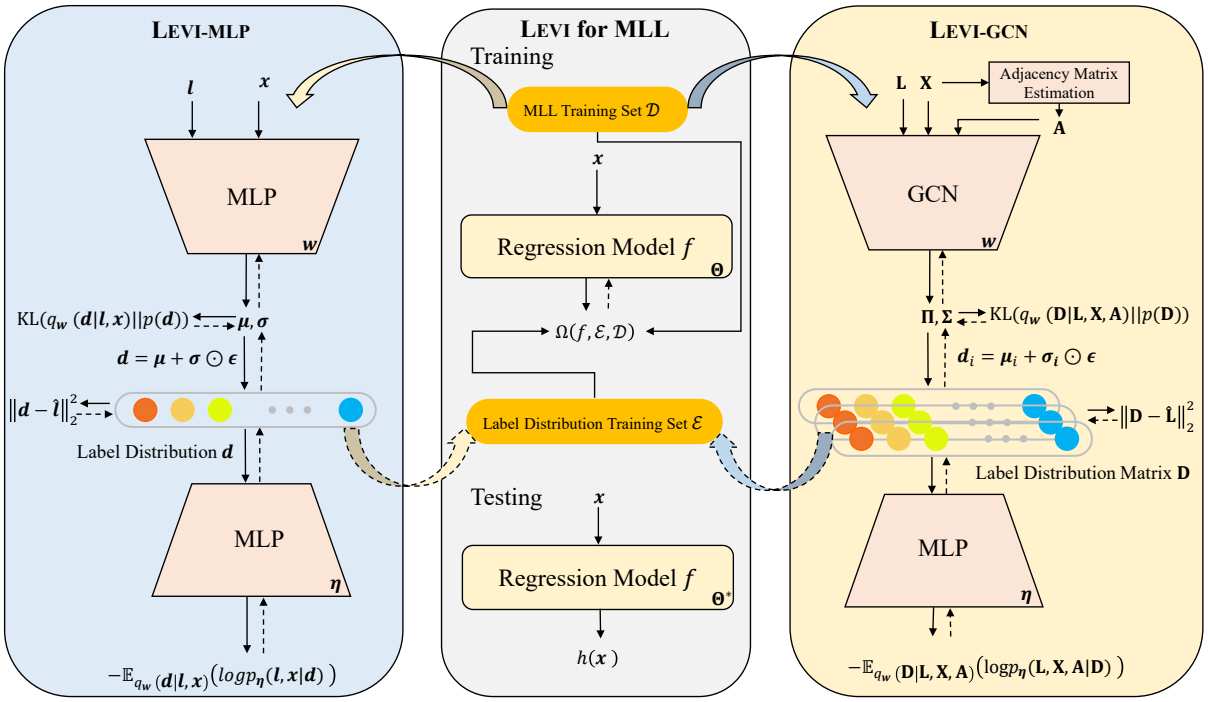


Fig. 2: The framework of the proposed methods. LEVI-MLP and LEVI-GCN are two LE approaches, where the inference model $\boldsymbol{w}$ is respectively instantiated by MLP and GCN to recover the label distributions from the training data. Then, the MLL training set $\mathcal{D}$ could be transformed into the label distribution training set $\mathcal{E}$, which induces the regression model to deal with multi-label learning. The black solid lines denote the forward process, and the black dotted lines mark the gradient flow.

along with a specialized objective function.

3 The Proposed Methods

First of all, the main notations used in this paper are listed as follows. $\boldsymbol{x}$ denotes the instance variable and $\boldsymbol{x}_i$ denotes the particular i-th instance. y denotes the class label and y_j denotes the particular j-th class label. The logical label vector corresponding to $\boldsymbol{x}_i$ is denoted by $\boldsymbol{l}_i = (l_{\boldsymbol{x}_i}^{y_1}, l_{\boldsymbol{x}_i}^{y_2}, ..., l_{\boldsymbol{x}_i}^{y_c})^\top$, where c is the number of labels. The description degree of y to $\boldsymbol{x}$ is denoted by $d_{\boldsymbol{x}}^{y}$, and the label distribution of $\boldsymbol{x}_i$ is denoted by $\boldsymbol{d}_i = (d_{\boldsymbol{x}_i}^{y_1}, d_{\boldsymbol{x}_i}^{y_2}, ..., d_{\boldsymbol{x}_i}^{y_c})^\top$. $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$, $\mathbf{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_n]$ and $\mathbf{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n]$ are feature matrix, logical label matrix and label distribution matrix, respectively, where n is the number of samples.

In the section, we firstly adopt the generative models of the label distribution and deduce the variational lower bound for LE. Then we instantiate the generative models as MLP and GCN [24] and propose LEVI-MLP and LEVI-GCN for LE. At last, we train the MLL predictive model via leveraging the recovered label distributions by LEVI-MLP and LEVI-GCN. The framework of the proposed method is shown in Fig. 2.

3.1 Variational Lower Bound

Given the difficulty and cost of quantifying the label distributions, people instead choose simplifying the supervised information by the logical labels. Therefore, the logical label vector $\boldsymbol{l}$ and the instance $\boldsymbol{x}$ can be treated as observed vectors, and the label distribution $\boldsymbol{d}$ is treated as a latent vector. Then the problem of label enhancement could be solved if the posterior density $p(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})$ is obtained. As computation of the exact posterior density $p(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})$ is intractable, we employ a fixed-form density $q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})$ to approximate the true posterior $p(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})$. By following the Variational Bayes techniques, we derive a lower bound which could ensure that the $q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})$ is as close as possible to $p(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})$.

We begin with the definition of Kullback-Leibler divergence (KL divergence) between $p(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})$ and $q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})$:

$$\begin{aligned}\mathrm{KL}[q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})||p(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})] = \mathbb{E}_{q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})}[\log q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x}) \\ - \log p(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})].\end{aligned} \tag{1}$$

Applying Bayes rule:

$$\begin{aligned}\mathrm{KL}[q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})||p(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})] = \mathbb{E}_{q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})}[\log q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x}) - \\ \log p(\boldsymbol{l}, \boldsymbol{x}|\boldsymbol{d}) - \log p(\boldsymbol{d}) + \log p(\boldsymbol{l}, \boldsymbol{x})].\end{aligned} \tag{2}$$

Here, $\log p(\boldsymbol{l}, \boldsymbol{x})$ comes out of the expectation because it does not depend on $\boldsymbol{d}$:

$$\begin{aligned}\mathrm{KL}[q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})||p(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})] = \mathrm{KL}[q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})||p(\boldsymbol{d})] \\ + \log p(\boldsymbol{l}, \boldsymbol{x}) - \mathbb{E}_{q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})}[\log p(\boldsymbol{l}, \boldsymbol{x}|\boldsymbol{d})].\end{aligned} \tag{3}$$

Since this KL-divergence is non-negative, we have :

$$\begin{aligned}\log p(\boldsymbol{l}, \boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})}[\log p(\boldsymbol{l}, \boldsymbol{x}|\boldsymbol{d})] \\ - \mathrm{KL}[q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})||p(\boldsymbol{d})].\end{aligned} \tag{4}$$

Therefore, label enhancement is the process which aims to maximize the lower bound of the joint probability density $p(\boldsymbol{l}, \boldsymbol{x})$ by recovering the optimal label distribution $\boldsymbol{d}$. We construct the approximate posterior density $q(\boldsymbol{d}|\boldsymbol{l}, \boldsymbol{x})$ as an

inference model, which is efficient variational inference [22], [35]. The parameters of $q(\boldsymbol{d}|\boldsymbol{l},\boldsymbol{x})$ and $p(\boldsymbol{l},\boldsymbol{x}|\boldsymbol{d})$ are modeled with $\boldsymbol{w}$ and $\boldsymbol{\eta}$, respectively. Then, the Evidence Lower Bound (ELBO) is written as

$$\begin{aligned}\mathcal{L}(\boldsymbol{x},\boldsymbol{l};\boldsymbol{w},\boldsymbol{\eta}) = \mathbb{E}_{q_{\boldsymbol{w}}(\boldsymbol{d}|\boldsymbol{l},\boldsymbol{x})}[\log p_{\boldsymbol{\eta}}(\boldsymbol{l},\boldsymbol{x}|\boldsymbol{d})] \\ -\mathrm{KL}[q_{\boldsymbol{w}}(\boldsymbol{d}|\boldsymbol{l},\boldsymbol{x})||p(\boldsymbol{d})].\end{aligned} \tag{5}$$

The bound in Eq. (5) provides a unified objective function for optimization of $\boldsymbol{w}$ and $\boldsymbol{\eta}$.

3.2 LEVI for Label Enhancement

In this paper, two label enhancement approaches, i.e., LEVI-MLP and LEVI-GCN are developed by handling the ELBO in Eq. (5) with different models. LEVI-MLP directly employs the multi-layer perceptron (MLP) to model the parameters of the approximate posterior density, which is simple yet efficient. Besides, LEVI-GCN is proposed by further considering the topological information of the feature space, which leads to adopting a graph convolutional network (GCN) [23] as the model of the approximate posterior density in Eq. (5).

3.2.1 LEVI-MLP

By expanding the label distribution into $\boldsymbol{d} \in R^c$, we assume that the prior over the latent label distribution is the centered isotropic multivariate Gaussian $p(\boldsymbol{d}) = \mathcal{N}(\boldsymbol{0},\mathbf{I})$. We let the variational approximate posterior be a multivariate Gaussian with a diagonal covariance structure $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\sigma}^2\mathbf{I})$. Here $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the c-dimensional mean and the standard deviation vectors, which are the outputs of a MLP with two hidden layers. Then the KL divergence in Eq. (5) can be computed:

$$\begin{aligned}\mathrm{KL}[q_{\boldsymbol{w}}(\boldsymbol{d}|\boldsymbol{l},\boldsymbol{x})||p(\boldsymbol{d})] = \frac{1}{2}\Big(\boldsymbol{\mu}^\top\boldsymbol{\mu} + \mathrm{tr}\left(\boldsymbol{\sigma}^2\mathbf{I}\right) - c \\ - \log|\boldsymbol{\sigma}^2\mathbf{I}|\Big).\end{aligned} \tag{6}$$

As there is a factorized form $p(\boldsymbol{l},\boldsymbol{x}|\boldsymbol{d}) = p(\boldsymbol{l}|\boldsymbol{x},\boldsymbol{d})p(\boldsymbol{x}|\boldsymbol{d})$, we let $p(\boldsymbol{l}|\boldsymbol{x},\boldsymbol{d})$ be a multivariate Bernoulli with probabilities $\boldsymbol{\tau}$ and $p(\boldsymbol{x}|\boldsymbol{d})$ be a multivariate Gaussian with means $\boldsymbol{\rho}$. Then the first part of Eq. (5) can be computed:

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{w}}(\boldsymbol{d}|\boldsymbol{l},\boldsymbol{x})}[\log p_{\boldsymbol{\eta}}(\boldsymbol{l},\boldsymbol{x}|\boldsymbol{d})] = \frac{1}{J}\sum_{j=1}^{J}\Big(-\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{\rho}^{(j)}\|_2^2 \\ +\boldsymbol{l}^\top\log\boldsymbol{\tau}^{(j)} + (\mathbf{1}-\boldsymbol{l})^\top\log\left(\mathbf{1}-\boldsymbol{\tau}^{(j)}\right)\Big),\end{aligned} \tag{7}$$

Here, to simplify the observation model, MC sampling [24] is employed in Eq. (9) during the training process, where $\boldsymbol{\tau}^{(j)}$ and $\boldsymbol{\rho}^{(j)}$ are computed from the j-th sampled $\boldsymbol{d}^{(j)}$ with the MLP parameterized by $\boldsymbol{\eta}$ and J is the sampling number. In order to move the sampling to an input layer, the reparameterization trick [35] is employed to sample $\boldsymbol{d}$ by:

$$\boldsymbol{d}^{(j)} = \boldsymbol{\mu} + \boldsymbol{\sigma}\odot\boldsymbol{\epsilon}^{(j)}, \tag{8}$$

where $\boldsymbol{\epsilon}^{(j)} \sim \mathcal{N}(\boldsymbol{0},\mathbf{I})$. In this case, Eq. (7) can be differentiated.

As the label distributions inherit relevance and irrelevance from the initial label vectors [49], we add the least squares for the label distribution and the initial label vectors into the objective function. The -1/1 label vector $\hat{\boldsymbol{l}} = [\hat{l}_{\boldsymbol{x}_i}^{y_1}, \hat{l}_{\boldsymbol{x}_i}^{y_2}, \ldots, \hat{l}_{\boldsymbol{x}_i}^{y_c}]^\top$ of each $\boldsymbol{x}_i$ is utilized in the least squares:

$$\forall_{j=0}^{c}: \quad \hat{l}_{\boldsymbol{x}_i}^{y_j} = \begin{cases} 1, & \text{if } y_j \in Y_i \\ -1, & \text{if } y_j \notin Y_i \end{cases} \tag{9}$$

where Y_i represents the relevant label set of $\boldsymbol{x}_i$. Then, we formulate the label enhancement problem into an optimization framework to yield the target function for minimization:

$$\begin{aligned}T(\boldsymbol{\eta},\boldsymbol{w}) = \sum_{i=1}^{n}\Bigg(\frac{1}{J}\sum_{j=1}^{J}\Big(\frac{1}{2}\|\boldsymbol{x}_i-\boldsymbol{\rho}_i^{(j)}\|_2^2 \\ +\lambda\|\boldsymbol{d}_i^{(j)}-\hat{\boldsymbol{l}}_i\|_2^2 - \boldsymbol{l}_i^\top\log\boldsymbol{\tau}_i^{(j)} \\ -(\mathbf{1}-\boldsymbol{l}_i)^\top\log\left(\mathbf{1}-\boldsymbol{\tau}_i^{(j)}\right)\Big) \\ +\frac{1}{2}\Big(\boldsymbol{\mu}_i^\top\boldsymbol{\mu}_i + \mathrm{tr}\left(\boldsymbol{\sigma}_i^2\mathbf{I}\right) - c - \log|\boldsymbol{\sigma}_i^2\mathbf{I}|\Big)\Bigg),\end{aligned} \tag{10}$$

where λ is a hyper-parameter. Stochastic gradient descent is utilized for the optimization. When $\boldsymbol{w}$ and $\boldsymbol{\eta}$ are determined, the label distribution $\boldsymbol{d}_i$ of each instance $\boldsymbol{x}_i$ can be sampled from the posterior $\boldsymbol{d}_i \sim q_{\boldsymbol{w}}(\boldsymbol{d}|\boldsymbol{l}_i,\boldsymbol{x}_i)$. In order to make the output of LEVI-MLP deterministic rather than stochastic, we let the output label distribution be equal to the mean of the variational approximate posterior in the experiments.

In the training procedure of LEVI-MLP, we first initialized the reference model $\boldsymbol{w}$ and the observation model $\boldsymbol{\eta}$. In each epoch, we calculated the label distribution $\boldsymbol{d}_i$ corresponding to each example $\boldsymbol{x}_i$ by Eq. (8) and updated $\boldsymbol{w}$ and $\boldsymbol{\eta}$ with back-propagation and forward computation by Eq. (10) on each mini-batch. Then, we transformed the MLL training set $\mathcal{D}$ into the label distribution training set $\mathcal{E}$

Algorithm 1 LEVI-MLP Algorithm

Input: The MLL training set $\mathcal{D} = \{(\boldsymbol{x}_i,\boldsymbol{l}_i)\}_{i=1}^n$, epoch T and iteration I;

```
Initialize the reference model w and the observation model η;
for t = 1, ..., T do
    Shuffle training set D = {(x_i, l_i)}_{i=1}^n into I mini-batches;
    for k = 1, ..., I do
        Calculate the label distribution d_i corresponding to each example x_i by Eq. (8);
        Update w and η with back-propagation and forward computation by Eq. (10);
    end for
end for
Obtain the label distributions d_i for each example x_i and generate the label distribution training set E = {(x_i, d_i)}_{i=1}^n;
Initialize the predictive model Θ^(0), t = 0;
repeat
    Calculate Θ^(s) via Eq. (30);
    Update Θ^(t+1) via line searching with Θ^(t) and Θ^(s);
    t = t + 1;
until convergence
```

Output: The predictive model $\boldsymbol{\Theta}$.

and induced the regression model to deal with multi-label learning as shown in Fig. 2. The algorithmic description of LEVI-MLP is shown in Algorithm 1.

3.2.2 LEVI-GCN

As recovering the label distribution could benefit from topological structure of the feature space in the LE process [49], LEVI-GCN further adopts a GCN to instantiate the inference model, which could explicitly leverage the topological structure of the feature space and naturally integrate the topological structure into the ELBO in Eq. (5) to recover the label distributions.

The topological structure of the feature space can be represented by the affinity graph $G = (V, E, \mathbf{P})$. Here, $V = \{\boldsymbol{x}_i \mid 1 \leq i \leq n\}$ corresponds to the vertex set consisting of feature vectors, $E = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid 1 \leq i \neq j \leq n\}$ corresponds to the edge set. Intuitively, the weight matrix $\mathbf{P} = [p_{ij}]_{n\times n}$ encodes the relationships among all training examples, where each weight p_{ij} reflects the influence of $\boldsymbol{x}_i$ over $\boldsymbol{x}_j$. Therefore, $\mathbf{P}$ could be estimated via modeling the relationship between one example and all the other examples via the reconstruction of each instance.

For each instance $\boldsymbol{x}_i$, LEVI-GCN aims to reconstruct $\boldsymbol{x}_i$ from all the other instances in the training set. The weight matrix $\mathbf{P} = [p_{ij}]_{n\times n}$ can be optimized by solving the following reconstruction problem [58]:

$$\min_{\bar{\boldsymbol{p}}_i} \left\|\bar{\mathbf{X}}_i\bar{\boldsymbol{p}}_i - \boldsymbol{x}_i\right\|_2^2 + \nu \left\|\bar{\boldsymbol{p}}_i\right\|_1, \tag{11}$$

where $\bar{\boldsymbol{p}}_i = [p_{1,i}, \ldots, p_{i-1,i}, p_{i+1,i}, \ldots, p_{n,i}]^\top$, $\bar{\mathbf{X}}_i = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_n]$, and ν is a tradeoff parameter. To deal with large datasets, we would only reconstruct $\boldsymbol{x}_i$ from its K-nearest neighbors. For $\boldsymbol{x}_i$, $p_{ji} = 0$ unless $\boldsymbol{x}_j$ is one of $\boldsymbol{x}_i$'s K-nearest neighbors. In the reconstruction problem in Eq. (11), we only consider its $(k-1)$-dimensional weight vector $\bar{\boldsymbol{p}}_i$ and the matrix $\bar{\mathbf{X}}_i$ constituted by $(k-1)$ vectors, and then Eq. (11) could be solved efficiently.

The optimization problem in Eq. (11) can be solved as a series of minimization problems by ADMM technique [17]:

$$\begin{aligned} L\left(\boldsymbol{p}_i, \boldsymbol{z}_i, \boldsymbol{\zeta}_i\right) =& \frac{1}{2}\left\|\mathbf{X}_i\boldsymbol{p}_i - \boldsymbol{x}_i\right\|_2^2 + \nu\left\|\boldsymbol{z}_i\right\|_1 \\ &+\boldsymbol{\zeta}_i^\top\left(\boldsymbol{p}_i - \boldsymbol{z}_i\right) + \frac{1}{2}\left\|\boldsymbol{v}_i - \boldsymbol{z}_i\right\|_2^2. \end{aligned} \tag{12}$$

The minimization of $\boldsymbol{p}_i$, $\boldsymbol{z}_i$ and $\boldsymbol{\zeta}_i$ can be conducted by the scaled ADMM iterations [58]. By solving the reconstruction problem of Eq. (11) for each instance $\boldsymbol{x}_i$, we instantiate $\bar{\mathbf{P}}$ with $\bar{\boldsymbol{p}}$ and zero diagonal elements. Then the symmetrical weight matrix $\mathbf{P} = \frac{1}{2}(\bar{\mathbf{P}} + \bar{\mathbf{P}}^\top)$.

LEVI-GCN employs Cantelli's inequality-based outlier thresholding [30] to generate a very sparse adjacency matrix $\mathbf{A} = [a_{ij}]_{n\times n}$:

$$\forall_{i\neq j}: \quad a_{ij} = \begin{cases} 1, & \text{if } p_{ij} \geq \bar{\mu} + \bar{\delta} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $\bar{\mu}$ and $\bar{\delta}$ are the expected value and variance of all the elements in $\mathbf{P}$ except zero diagonal elements, respectively. The diagonal elements of $\mathbf{A}$ are set to 1. Then, the topological information of the feature space is utilized by adding $\mathbf{A}$ into Eq. (5):

$$\begin{aligned} \mathcal{L}(\mathbf{L}, \mathbf{X}, \mathbf{A}; \boldsymbol{\eta}, \boldsymbol{w}) = & \mathbb{E}_{q_{\boldsymbol{w}}(\mathbf{D}|\mathbf{L},\mathbf{X},\mathbf{A})}[\log p_{\boldsymbol{\eta}}(\mathbf{L}, \mathbf{X}, \mathbf{A}|\mathbf{D})] \\ & -\mathrm{KL}[q_{\boldsymbol{w}}(\mathbf{D}|\mathbf{L}, \mathbf{X}, \mathbf{A})||p(\mathbf{D})]. \end{aligned} \tag{14}$$

LEVI-GCN assumes that the prior over the latent label distribution is the centered isotropic multivariate Gaussian:

$$p(\mathbf{D}) = \prod_i p\left(\boldsymbol{d}_i\right) = \prod_{i=1}^n \mathcal{N}\left(\boldsymbol{d}_i \mid \mathbf{0}, \mathbf{I}\right) \tag{15}$$

In addition, we let the variational approximate posterior be the product of each multivariate Gaussian with a diagonal covariance structure, where the mean and standard deviation matrix of each multivariate Gaussian, $\boldsymbol{\Pi} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_n]$ and $\boldsymbol{\Sigma} = [\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n]$, are outputs of a two-layer GCN [24]:

$$\begin{aligned} q(\mathbf{D} \mid \mathbf{L}, \mathbf{X}, \mathbf{A}) &= \prod_{i=1}^n q\left(\boldsymbol{d}_i \mid \mathbf{L}, \mathbf{X}, \mathbf{A}\right) \\ &= \prod_{i=1}^n \mathcal{N}\left(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\mathbf{I}\right). \end{aligned} \tag{16}$$

Here, the two-layer GCN parameterized by $\boldsymbol{w}$ is defined as $\mathrm{GCN}(\mathbf{L}, \mathbf{X}, \mathbf{A}) = \tilde{\mathbf{A}}\,\mathrm{ReLU}\left(\tilde{\mathbf{A}}\mathbf{Z}\mathbf{W}_0\right)\mathbf{W}_1$, with $\mathbf{Z} = [\mathbf{X}; \mathbf{L}]$ and weight $\mathbf{W}_0$, $\mathbf{W}_1$. $\tilde{\mathbf{A}} = \hat{\mathbf{A}}^{-\frac{1}{2}}\mathbf{A}\hat{\mathbf{A}}^{-\frac{1}{2}}$ is the symmetrically normalized weight matrix, where $\hat{\mathbf{A}}$ is the degree matrix of $\mathbf{A}$. Then, the KL divergence in the Eq. (14) can be computed:

$$\begin{aligned} \mathrm{KL}[q_{\boldsymbol{w}}(\mathbf{D}|\mathbf{L}, \mathbf{X}, \mathbf{A})||p(\mathbf{D})] = & \frac{1}{2}\sum_{i=1}^n \Big(\mathrm{tr}(\boldsymbol{\sigma}_i^2\mathbf{I}) + \boldsymbol{\mu}_i^\top\boldsymbol{\mu}_i \\ & -c - \log\left|\boldsymbol{\sigma}_i^2\mathbf{I}\right| \Big). \end{aligned} \tag{17}$$

As there is a factorized form $p(\mathbf{L}, \mathbf{X}, \mathbf{A}|\mathbf{D}) = p(\mathbf{L}|\mathbf{X}, \mathbf{A}, \mathbf{D})p(\mathbf{X}|\mathbf{A}, \mathbf{D})p(\mathbf{A}|\mathbf{D})$. Then our generative model is given by

$$\begin{aligned} p(\mathbf{L} \mid \mathbf{X}, \mathbf{A}, \mathbf{D}) &= \prod_{i=1}^n p\left(\boldsymbol{l}_i \mid \mathbf{X}, \mathbf{A}, \mathbf{D}\right), \\ p(\mathbf{X} \mid \mathbf{A}, \mathbf{D}) &= \prod_{i=1}^n p\left(\boldsymbol{x}_i \mid \mathbf{A}, \mathbf{D}\right), \\ p(\mathbf{A} \mid \mathbf{D}) &= \prod_{i=1}^n \prod_{j=1}^n p\left(a_{ij} \mid \boldsymbol{d}_i, \boldsymbol{d}_j\right). \end{aligned} \tag{18}$$

We further assume that $p\left(\boldsymbol{l}_i|\mathbf{X}, \mathbf{A}, \mathbf{D}\right)$ is a multivariate Bernoulli with probabilities $\boldsymbol{\tau}_i$ and $p(\boldsymbol{x}_i|\mathbf{A}, \boldsymbol{D})$ is a multivariate Gaussian with means $\boldsymbol{\rho}_i$. Then the first part of Eq. (14) can be computed:

$$\begin{aligned} &\mathbb{E}_{q_{\boldsymbol{w}}(\mathbf{D}|\mathbf{L},\mathbf{X},\mathbf{A})}[\log p_{\boldsymbol{\eta}}(\mathbf{L}, \mathbf{X}, \mathbf{A}|\mathbf{D})] = \frac{1}{J}\sum_{j=1}^J \\ &\Big(\mathrm{tr}\left(\mathbf{L}^\top \log \mathbf{T}^{(j)}\right) + \mathrm{tr}\left((\mathbf{I}-\mathbf{L})^\top \log\left(\mathbf{I} - \mathbf{T}^{(j)}\right)\right) \\ &-\frac{1}{2}\|\mathbf{X} - \mathbf{E}^{(j)}\|_F^2 - \|\mathbf{A} - S\left(\mathbf{D}^{(j)}\mathbf{D}^{(j)\top}\right)\|_F^2 \Big). \end{aligned} \tag{19}$$

In order to simplify the observation model, $\mathbf{T}^{(j)} = [\boldsymbol{\tau}_1^{(j)}, \boldsymbol{\tau}_2^{(j)}, \ldots, \boldsymbol{\tau}_n^{(j)}]$ and $\mathbf{E}^{(j)} = [\boldsymbol{\rho}_1^{(j)}, \boldsymbol{\rho}_2^{(j)}, \ldots, \boldsymbol{\rho}_n^{(j)}]$ are computed from j-th sampling $\boldsymbol{D}^{(j)}$ with the MLP parameterized by $\boldsymbol{\eta}$. $S(\cdot)$ is the logistic sigmoid function.

Finally, we formulate the label enhancement problem into an optimization framework with the least squares of the

label distribution $\boldsymbol{d}_i$ and the logical label vectors $\hat{\boldsymbol{l}}_i$, which yields the target function for minimization:

$$\begin{aligned} T(\boldsymbol{\eta}, \boldsymbol{w}) = \frac{1}{J}\sum_{j=1}^{J}\Bigg(&\frac{1}{2}\|\mathbf{X}-\mathbf{E}\|_F^2 + \lambda\|\mathbf{D}^{(j)}-\hat{\mathbf{L}}\|_F^2 \\ &- \operatorname{tr}\left(\mathbf{L}^\top \log \mathbf{T}\right) - \operatorname{tr}\left((\mathbf{I}-\mathbf{L})^\top \log(\mathbf{I}-\mathbf{T})\right) \\ &+ \|\mathbf{A} - S\left(\mathbf{D}^{(j)}\mathbf{D}^{(j)\top}\right)\|_F^2\Bigg) + \frac{1}{2}\sum_{i=1}^{n}\Bigg(\boldsymbol{\mu}_i^\top\boldsymbol{\mu}_i \\ &+ \operatorname{tr}(\boldsymbol{\sigma}_i^2\mathbf{I}) - c - \log|\boldsymbol{\sigma}_i^2\mathbf{I}|\Bigg), \end{aligned} \tag{20}$$

where λ is a tradeoff parameter and $\hat{\mathbf{L}} = [\hat{\boldsymbol{l}}_1, \hat{\boldsymbol{l}}_2, \ldots, \hat{\boldsymbol{l}}_n]$.

When $\boldsymbol{\eta}$ and $\boldsymbol{w}$ are determined, the label distribution matrix $\mathbf{D}$ can be sampled from the approximate posterior $\mathbf{D} \sim q(\mathbf{D}|\mathbf{L}, \mathbf{X}, \mathbf{A})$. In order to make the output of LEVI-GCN deterministic rather than stochastic, we let the output label distributions be equal to the means of the approximate posterior in the experiments.

LEVI-GCN models the topological information by employing the adjacency matrix $\mathbf{A} = [a_{ij}]_{n\times n}$ which encodes the relationships among all training examples and leverages the topological information via Eq. (14) with the added adjacency matrix. Comparing to LEVI-GCN, LEVI-MLP only considers the relationship between instance $\boldsymbol{x}$ and the corresponding label $\boldsymbol{l}$ via Eq. (5) so that it cannot explicitly capture the topological information of the feature space.

In the training procedure of LEVI-GCN, we first initialize the reference model $\boldsymbol{w}$, the observation model $\boldsymbol{\eta}$ and the weight matrix $\mathbf{P}$. Then the adjacency matrix $\mathbf{A}$ was obtained by calculating Eq. (13) with weight matrix $\mathbf{P}$ generated by solving the minimization problem in Eq. (11) via ADMM technique. In each epoch, we calculated the label distribution $\boldsymbol{d}_i$ corresponding to each example $\boldsymbol{x}_i$ by Eq. (16) and updated $\boldsymbol{w}$ and $\boldsymbol{\eta}$ with back-propagation and forward computation by Eq. (20) on each mini-batch. Then, we transformed the MLL training set $\mathcal{D}$ into the label distribution training set $\mathcal{E}$ and induced the regression model to deal with multi-label learning as shown in Fig. 2. The algorithmic description of LEVI-GCN is shown in Algorithm 2.

Algorithm 2 LEVI-GCN Algorithm

Input: The MLL training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{l}_i)\}_{i=1}^n$, epoch T and iteration I;
Initialize the reference model $\boldsymbol{w}$, the observation model $\boldsymbol{\eta}$, and the weight matrix $\mathbf{P}$;
Obtain the weight matrix $\mathbf{P}$ by solving the minimization problem in Eq. (11) via ADMM technique.
Calculate the adjacency matrix $\mathbf{A}$ via Eq. (13);
for $t = 1, \ldots, T$ **do**
  Shuffle training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{l}_i)\}_{i=1}^n$ into I mini-batches;
  for $k = 1, \ldots, I$ **do**
    Calculate the label distribution $\boldsymbol{d}_i$ corresponding to each example $\boldsymbol{x}_i$ by Eq. (16);
    Update $\boldsymbol{w}$ and $\boldsymbol{\eta}$ with back-propagation and forward computation by Eq. (20);
  end for
end for
Obtain the label distributions $\boldsymbol{d}_i$ for each example $\boldsymbol{x}_i$ and generate the label distribution training set $\mathcal{E} = \{(\boldsymbol{x}_i, \boldsymbol{d}_i)\}_{i=1}^n$;
Initialize the predictive model $\boldsymbol{\Theta}^{(0)}$, $t = 0$;
repeat
  Calculate $\boldsymbol{\Theta}^{(s)}$ via Eq. (30);
  Update $\boldsymbol{\Theta}^{(t+1)}$ via line searching with $\boldsymbol{\Theta}^{(t)}$ and $\boldsymbol{\Theta}^{(s)}$;
  $t = t + 1$;
until convergence
Output: The predictive model $\boldsymbol{\Theta}$.

3.3 LEVI for Multi-Label Learning

When the label distribution $\boldsymbol{d}_i$ of each $\boldsymbol{x}_i$ is recovered by LEVI-MLP or LEVI-GCN, the multi-label training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{l}_i) \mid 1 \le i \le n\}$ can be transformed into the label distribution training set $\mathcal{E} = \{(\boldsymbol{x}_i, \boldsymbol{d}_i) \mid 1 \le i \le n\}$.

As $\boldsymbol{d}_i$ is real-valued, multi-output support vector regression (MSVR) [5], [31] is employed to handle this case, where the kernel regression model is used to parametrize the multi-label predictor:

$$\begin{aligned} \forall_{j=1}^c: \quad f(y_j|\boldsymbol{x}_i, \boldsymbol{\Theta}, \boldsymbol{b}) &= \hat{\boldsymbol{\theta}}_j\varphi(\boldsymbol{x}_i) + b_j \\ &= \boldsymbol{\theta}_j^\top \boldsymbol{\phi}_i. \end{aligned} \tag{21}$$

Here, $\boldsymbol{\theta}_j = [\hat{\boldsymbol{\theta}}_j^\top, b_j]^\top$, $\boldsymbol{\phi}_i = [\varphi(\boldsymbol{x}_i)^\top, 1]^\top$, $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_c]$, and $\varphi(\boldsymbol{x}_i)$ is a nonlinear transformation of $\boldsymbol{x}_i$ to a higher dimensional feature space. According to the representer theorem [39], a learning problem would be represented as a linear combination of the training data in the feature space under fairly general conditions, i.e. $\boldsymbol{\theta}^j = \sum_i \boldsymbol{\eta}^j\varphi(\boldsymbol{x}_i)$. By replacing this expression into final objective function, the inner product $< \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) >$ could be generated, and then the kernel trick can be applied.

As the kernel regression model could handle the nonlinear mapping problem and possess the higher interpretability than MLP, we employ the kernel regression model as the predictive model for MLL in this section. This is different from the model selection that MLP is employed to model the parameters of the fixed-form density $q(\boldsymbol{d}|\boldsymbol{x}, \boldsymbol{l})$ in Section 3.2 since the high-capacity MLP will hopefully make variational posterior density approximate the true posterior density [8] and is widely-used in variational reference [24].

The multi-label predictive model is induced by optimizing the following objective function:

$$\Omega(f, \mathcal{E}, \mathcal{D}) = \frac{1}{2}\|\boldsymbol{\Theta}\|_F^2 + \beta\Omega_1(f, \mathcal{E}) + \gamma\Omega_2(f, \mathcal{D}). \tag{22}$$

The first term of $\Omega(f, \mathcal{E}, \mathcal{D})$ controls the complexity of the induced model. Besides, $\Omega_1(f, \mathcal{E})$ concerns the distance between the predictions and the label distribution, and $\Omega_2(f, \mathcal{D})$ is employed to keep the sign consistency of the prediction and the ground-truth.

$\Omega_1(f, \mathcal{E})$ is defined to yield a single support vector:

$$\Omega_1(f, \mathcal{E}) = \sum_{i=1}^{n} V(r_i) \tag{23}$$

Here, $r_i = \|\boldsymbol{e}_i\| = \sqrt{\boldsymbol{e}_i^\top \boldsymbol{e}_i}$, $\boldsymbol{e}_i = \boldsymbol{d}_i - \boldsymbol{\Theta}^\top\varphi(\boldsymbol{x}_i) - \boldsymbol{b}$. $V(z) = (z - \varepsilon)^2$ if $z \ge \varepsilon$, and $V(z) = 0$ otherwise. This term could

generate an insensitive zone determined by ε around the estimation, i.e., the loss of r less than ε will be ignored.

$\Omega_2(f, \mathcal{D})$ is employed to keep the signs of the prediction and the ground-truth label $\hat{\boldsymbol{l}}$ consistent:

$$\begin{aligned}\Omega_2(f, \mathcal{D}) &= -\sum_{i=1}^{n}\sum_{j=1}^{c} \hat{l}^j_{\boldsymbol{x}_i} \boldsymbol{\theta}_j^\top \boldsymbol{\phi}_i \\ &= -\mathrm{tr}\left(\hat{\mathbf{L}}^\top \boldsymbol{\Theta}\boldsymbol{\Phi}\right),\end{aligned} \tag{24}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n]$ and $\hat{\mathbf{L}} = [\hat{\boldsymbol{l}}_1, \hat{\boldsymbol{l}}_2, \ldots, \hat{\boldsymbol{l}}_c]$.

To minimize the objective function $\Omega(f, \mathcal{E}, \mathcal{D})$, we choose to adopt an iterative quasi-Newton method called Iterative Re-Weighted Least Square (IRWLS) [32]. Firstly, $\Omega(f, \mathcal{E}, \mathcal{D})$ is approximated by its first order Taylor expansion at the solution of the current k-th iteration, denoted by $\boldsymbol{\Theta}^{(k)}$:

$$\tilde{V}(r_i) = V(r_i{}^{(k)}) + \left.\frac{dV}{dr}\right|_{r_i^{(k)}} \frac{(\boldsymbol{e}_i^{(k)})^\top}{r_i^{(k)}} \left(\boldsymbol{e}_i - \boldsymbol{e}_i^{(k)}\right), \tag{25}$$

where $\boldsymbol{e}_i^{(k)}$ and $r_i^{(k)}$ are calculated from $\boldsymbol{\Theta}^{(k)}$ and $\boldsymbol{b}^{(k)}$. Then a quadratic approximation is further constructed as

$$\begin{aligned}\bar{V}(r_i) =& V(r_i{}^{(k)}) + \left.\frac{dV(r_i)}{dr_i}\right|_{r_i^{(k)}} \frac{r_i^2 - (r_i^{(k)})^2}{2r_i^{(k)}} \\ =& \frac{1}{2}\xi_i r_i^2 + \tau,\end{aligned} \tag{26}$$

where

$$\xi_i = \frac{1}{r_i^{(k)}} \left.\frac{V(r_i)}{dr_i}\right|_{r_i^{(k)}} = \begin{cases} 0 & r_i^{(k)} < \varepsilon \\ \frac{2\left(r_i^{(k)} - \varepsilon\right)}{r_i^{(k)}} & r_i^{(k)} \geq \varepsilon, \end{cases} \tag{27}$$

and τ is a constant term that does not depend on either $\boldsymbol{\Theta}^{(k)}$ or $\boldsymbol{b}^{(k)}$. Combining Eq. (22), (24) and (26), we could obtain:

$$\begin{aligned}\bar{\Omega}(f, \mathcal{E}, \mathcal{D}) &= \frac{1}{2}\|\boldsymbol{\Theta}\|_F^2 + \frac{1}{2}\beta\sum_{i=1}^{n} a_i r_i^2 - \gamma \mathrm{tr}\left(\hat{\mathbf{L}}^\top \boldsymbol{\Theta}\boldsymbol{\Phi}\right) \\ &= \frac{1}{2}\|\boldsymbol{\Theta}\|_F^2 - \gamma \mathrm{tr}\left(\hat{\mathbf{L}}^\top \boldsymbol{\Theta}\boldsymbol{\Phi}\right) \\ &\quad + \frac{1}{2}\beta\left((\mathbf{D} - \boldsymbol{\Theta}^\top\boldsymbol{\Phi})\mathbf{H}(\mathbf{D} - \boldsymbol{\Theta}^\top\boldsymbol{\Phi})^\top\right).\end{aligned} \tag{28}$$

Here, $\mathbf{H} = [h_{ij}]_{n\times n}$ with $h_{ij} = \xi_i\delta_{ij}$ where δ_{ij} is the Kronecker's delta function. By making the corresponding gradient to be zero:

$$\nabla_{\boldsymbol{\Theta}} = \beta\boldsymbol{\Phi}\mathbf{H}\boldsymbol{\Phi}^\top\boldsymbol{\Theta} - \beta\boldsymbol{\Phi}\mathbf{H}\mathbf{D}^\top + \gamma\boldsymbol{\Phi}\hat{\mathbf{L}}^\top + \boldsymbol{\Theta} = \mathbf{0}, \tag{29}$$

the solution is obtained as

$$\boldsymbol{\Theta}^s = \left(\beta\boldsymbol{\Phi}\mathbf{H}\boldsymbol{\Phi}^\top + \mathbf{I}\right)^{-1}(\beta\boldsymbol{\Phi}\mathbf{H}\mathbf{D}^\top - \gamma\boldsymbol{\Phi}\hat{\mathbf{L}}^\top). \tag{30}$$

Then, the solution to the next iteration $\boldsymbol{\Theta}^{(k+1)}$ is obtained via a line search algorithm with $\boldsymbol{\Theta}^s$ and $\boldsymbol{\Theta}^{(k)}$. The algorithmic descriptions about the optimization processes of LEVI-MLP and LEVI-GCN for MLL are shown in Algorithm 1 and Algorithm 2, respectively.

Let $\boldsymbol{\Theta}^*$ be the resulting model after the whole iterative optimization process, the prediction is made on the output of unseen instance $\boldsymbol{x}$ with Eq. (21). Then, the predicted label set for $\boldsymbol{x}$ is determined as:

$$h(\boldsymbol{x}) = \{y_j \mid f(y_j|\boldsymbol{x}, \boldsymbol{\Theta}^*, \boldsymbol{b}^*) > 0, 1 \leq j \leq c\}. \tag{31}$$

TABLE 1: Statistics of the 14 datasets adopted in the label distribution recovery experiment

No.	Dataset	#Examples	#Features	#Labels
1	Artificial (Ar)	2601	3	3
2	SJAFFE (SJ)	213	243	6
3	Yeast-spoem (spoem)	2,465	24	2
4	Yeast-spo5 (spo5)	2,465	24	3
5	Yeast-dtt (dtt)	2,465	24	4
6	Yeast-cold (cold)	2,465	24	4
7	Yeast-heat (heat)	2,465	24	6
8	Yeast-spo (spo)	2,465	24	6
9	Yeast-diau (diau)	2,465	24	7
10	Yeast-elu (elu)	2,465	24	14
11	Yeast-cdc (cdc)	2,465	24	15
12	Yeast-alpha (alpha)	2,465	24	18
13	SBU_3DFE (3DFE)	2,500	243	6
14	Movie (Mov)	7,755	1,869	5

4 EXPERIMENTS

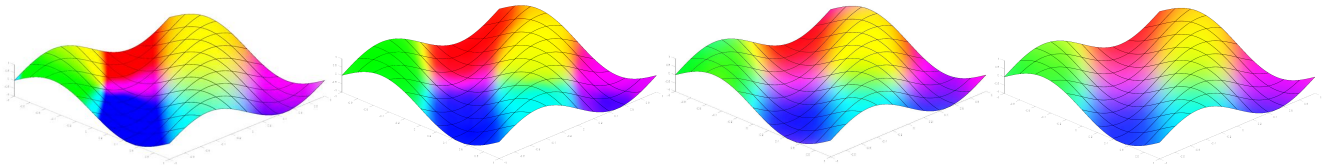
4.1 Label Distribution Recovery

In this experiment, the label distributions are recovered from the datasets with logical labels by LEVI-MLP, LEVI-GCN and other label enhancement algorithms, and then compared with the ground-truth label distributions in terms of six label distribution evaluation metrics.

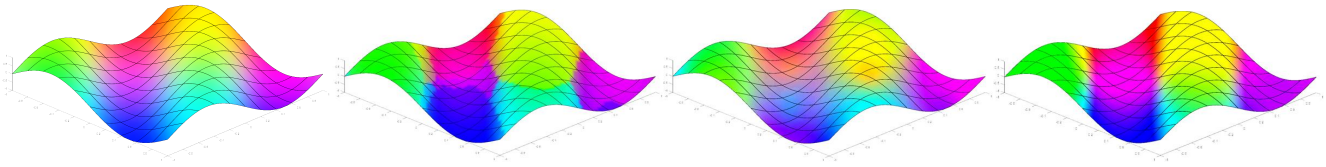
4.1.1 Datasets

There are in total one artificial dataset and 13 real-world label distribution datasets [14], whose basic statistics are given in Table 1. The datasets have been collected from several tasks and domains including `Yeast-spoem` to `Yeast-alpha` with phylogenetic profile vectors from the biological experiments on the budding yeast Saccharomyces cerevisiae, `SBU_3DFE` and `SJAFFE` with images from the facial expression estimation task, `Movie` with videos from the movie rating task, and `Artificial` generated from a certain manifold to show the results directly and visually.

- `Artificial` is generated to show the result of each label enhancement algorithm in a visual way. In this dataset, the examples are generated from a certain manifold to show the results directly and visually. The instance $\boldsymbol{x}$ is a three-dimensional vector with three class labels. The label distribution $\boldsymbol{d} = [d_{\boldsymbol{x}}^{y_1}, d_{\boldsymbol{x}}^{y_2}, d_{\boldsymbol{x}}^{y_3}]$ of $\boldsymbol{x} = [x_1, x_2, x_3]^\top$ is generated to deliberately make the description degree of one label depend on other labels [14].
- `Yeast-spoem` to `Yeast-alpha` are derived from the biological experiments on the budding yeast [9]. Each dataset records one biological experiment and contains 2,465 yeast genes represented by a phylogenetic profile vector. The discrete time points during one experiment constitute the labels in each dataset. The label distribution is constituted by the gene expression level at each time point.
- `SBU_3DFE` is a facial expression dataset which contains the basic emotions including sadness, happiness, fear, surprise, anger and disgust [53]. The level of emotional intensity (1 to 5) of each facial expression is annotated by twenty-three persons. The label distribution of each facial expression is constituted by the averaged intensities.



(a) Ground-Truth (b) LEVI-GCN (c) LEVI-MLP (d) LESC



(e) PLEML (f) LP (g) ML (h) GLLE

Fig. 3: The visualization of the ground-truth and the recovered label distributions (RGB colors) on the artificial dataset.

- `SJAFFE` is a facial expression dataset [29] which contains the same emotions as SBU_3DFE. Similarly, the level of emotional intensity is annotated by sixty persons and the label distribution of each the facial expression is constituted by averaged intensities.
- `Movie` is a movie dataset which contains 7,755 movies and 54,242,292 ratings from 478,656 different users [15]. The ratings are denoted from 1 to 5 stars (5 classes). The label distribution is calculated for each movie as the percentage of each rating level.

The discrete labels of each label distribution dataset are obtained via employing the most binarization method in LE [20], [43], [49], [59] as follows. For each training example, we select the greatest description degree $d_{\boldsymbol{x}}^{y_j}$ in the label distribution and set the corresponding class label y_j as the relevant label, i.e., $l_{\boldsymbol{x}}^{y_j} = 1$. Then, we calculated the sum of the description degrees corresponding to current relevant labels via $H = \sum_{y_j \in \mathcal{Y}^+} d_{\boldsymbol{x}}^{y_j}$, where $\mathcal{Y}^+$ denotes the set of the current relevant labels. We continually seek the greatest description degree among other labels excluded from $\mathcal{Y}^+$ and add the corresponding class label to $\mathcal{Y}^+$ until $H > T$, where T is a predefined threshold. Finally, we set all the labels in $\mathcal{Y}^+$ to 1 and the other labels to 0. We adopt the widely-used threshold $T = 0.5$ in the experiments [20], [43], [49], [59].

4.1.2 Evaluation Metrics

As suggested in [14], [49], we select four evaluation metrics including Chebyshev distance, Kullback-Leibler divergence, cosine coefficient, and intersection similarity, which belong to the Minkowski family, the Shannon's entropy family, the inner product family, and the intersection family, respectively. These metrics are significantly different in both syntax and semantics. The first two are distance measures and the last two are similarity measures.

- *Chebyshev distance* $\downarrow$
  $D_{Cheb} = \frac{1}{n}\sum_{i=1}^{n} \max_j |d_{\boldsymbol{x}_i}^{y_j} - \hat{d}_{\boldsymbol{x}_i}^{y_j}|$;
- *Kullback-Leibler divergence* $\downarrow$
  $D_{KL} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} d_{\boldsymbol{x}_i}^{y_j} \ln \frac{d_{\boldsymbol{x}_i}^{y_j}}{\hat{d}_{\boldsymbol{x}_i}^{y_j}}$;
- *Cosine coefficient* $\uparrow$
  $S_{Cos} = \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_{j=1}^{c} d_{\boldsymbol{x}_i}^{y_j} \hat{d}_{\boldsymbol{x}_i}^{y_j}}{\sqrt{\sum_{j=1}^{c} (d_{\boldsymbol{x}_i}^{y_j})^2}\sqrt{\sum_{j=1}^{c} (\hat{d}_{\boldsymbol{x}_i}^{y_j})^2}}$;
- *Intersection similarity* $\uparrow$
  $S_{Inter} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} \min(d_{\boldsymbol{x}_i}^{y_j}, \hat{d}_{\boldsymbol{x}_i}^{y_j})$.

Here, $\boldsymbol{d}_i = [d_{\boldsymbol{x}_i}^{y_1}, d_{\boldsymbol{x}_i}^{y_2}, \ldots, d_{\boldsymbol{x}_i}^{y_c}]$ denotes a real label distribution, $\hat{\boldsymbol{d}}_i = [\hat{d}_{\boldsymbol{x}_i}^{y_1}, \hat{d}_{\boldsymbol{x}_i}^{y_2}, \ldots, \hat{d}_{\boldsymbol{x}_i}^{y_c}]$ denotes a recovered label distribution. "$\downarrow$" indicates "the smaller the better", and "$\uparrow$" indicates "the larger the better".

4.1.3 Comparing Algorithms

Five baseline algorithms are utilized for comparative studies:

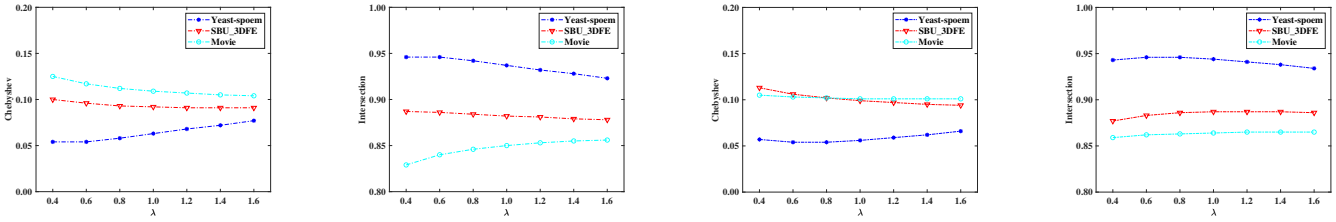
- LP [55] generates the label distributions via iterative label propagation technique [suggested configuration: trade-off hyper-parameter $\alpha = 0.5$].
- ML [19] estimates the label distributions via leveraging the feature manifold and the label manifold [suggested configuration: the number of neighbors $K = c + 1$].
- GLLE [52] adopts the graph laplacian with the topological structure of the feature space to generate the label distributions [suggested configuration: the hyper-parameters λ_1 and λ_2 are selected among $\{10^{-2}, 10^{-1}, ..., 100\}$].
- LESC [43] is a low-rank representation LE method via capturing the global relationships of samples and predicting the implicit label correlation [suggested configuration: the parameters λ_1 and λ_2 are selected among $\{10^{-4}, 10^{-3}, \ldots, 10\}$].
- PLEML [61] adopts the structural information between instances and the privileged information to recover label distributions [suggested configuration: the parameters λ_1 and λ_2 are selected among $\{2^{-4}, 2^{-3}, \ldots, 2^{8}\}$, $\gamma = 0.1$, and $C = 0.1$].

We employ three-layer MLP and two-layer GCN as the encoding models of LEVI-MLP and LEVI-GCN, respectively, and three-layer MLP as the decoding models of LEVI-MLP and LEVI-GCN. The numbers of hidden-layer nodes in MLP and GCN are set to 500. The hyper-parameter λ is set to 1. We use Adam as the optimizer and the learning rate and the weight decay are set to 1e-3 and 1e-5, respectively. Source code is available.[1]

1. https://github.com/palm-ml/LEVI

TABLE 2: Recovery results evaluated by six label distribution evaluation metrics. • and ○ denote the best and second best performance among all the approaches respectively.

Comparing algorithm	*Chebyshev distance* ↓												
	SJ	spoem	spo5	dtt	cold	heat	spo	diau	elu	cdc	alpha	3DFE	Mov
LP	0.107	0.163	0.114	0.128	0.137	0.086	0.090	0.099	0.044	0.042	0.040	0.123	0.161
ML	0.186	0.403	0.273	0.244	0.242	0.165	0.171	0.148	0.072	0.071	0.057	0.233	0.164
GLLE	0.087	0.088	0.099	0.052	0.066	0.049	0.062	0.053	0.023	0.022	0.020	0.126	0.122
LESC	0.069 •	0.087	0.092	0.043	0.056	0.046	0.060	0.042	0.019	0.019	0.015	0.122	0.121
PLEML	0.097	0.089	0.092	0.037 ○	0.054	0.044	0.060	0.042	0.017 ○	0.017	0.014	0.121 ○	0.166
LEVI-MLP	0.073 ○	0.063 ○	0.067 ○	0.034 •	0.051 ○	0.033 ○	0.045 ○	0.033 ○	0.012 •	0.015 ○	0.013 ○	0.092 •	0.109 •
LEVI-GCN	0.077	0.061 •	0.064 •	0.034 •	0.049 •	0.031 •	0.042 •	0.030 •	0.012 •	0.013 •	0.010 •	0.092 •	0.112 ○
Comparing algorithm	*Kullback-Leibler divergence* ↓												
	SJ	spoem	spo5	dtt	cold	heat	spo	diau	elu	cdc	alpha	3DFE	Mov
LP	0.077	0.067	0.042	0.103	0.103	0.089	0.084	0.127	0.109	0.111	0.121	0.105	0.177
ML	0.391	0.503	0.317	0.586	0.556	0.556	0.532	0.509	0.589	0.601	0.602	0.565	0.218
GLLE	0.050	0.027	0.034	0.013	0.019	0.017	0.029	0.027	0.013	0.014	0.013	0.069	0.123
LESC	0.029 •	0.027	0.032	0.009	0.015	0.016	0.027	0.017	0.009	0.010	0.008	0.064	0.120
PLEML	0.066	0.027	0.030	0.006 ○	0.014	0.013	0.027	0.016	0.007 ○	0.007	0.006 ○	0.064	0.170
LEVI-MLP	0.031 ○	0.013 ○	0.015 ○	0.005 •	0.011 ○	0.008 ○	0.014 ○	0.011 ○	0.005 •	0.006 ○	0.006 ○	0.042 ○	0.081 •
LEVI-GCN	0.029 •	0.012 •	0.014 •	0.005 •	0.010 •	0.007 •	0.013 •	0.009 •	0.005 •	0.005 •	0.004 •	0.041 •	0.084 ○
Comparing algorithm	*Cosine coefficient* ↑												
	SJ	spoem	spo5	dtt	cold	heat	spo	diau	elu	cdc	alpha	3DFE	Mov
LP	0.941	0.950	0.969	0.921	0.925	0.932	0.939	0.915	0.918	0.916	0.911	0.922	0.929
ML	0.857	0.815	0.884	0.763	0.784	0.783	0.803	0.803	0.763	0.759	0.756	0.815	0.919
GLLE	0.958	0.978 ○	0.971	0.988	0.982	0.984	0.974	0.975	0.987	0.987	0.987	0.927	0.936
LESC	0.973 •	0.978 ○	0.974	0.992	0.986	0.986	0.975 ○	0.985	0.991	0.991	0.992	0.932	0.937
PLEML	0.948	0.977	0.974	0.994 ○	0.987	0.987	0.975 ○	0.985	0.994 ○	0.993	0.995 ○	0.936 ○	0.882
LEVI-MLP	0.970	0.990 •	0.987 ○	0.995 •	0.990 ○	0.992 ○	0.988 •	0.990 ○	0.996 •	0.994 ○	0.995 ○	0.957 •	0.955 •
LEVI-GCN	0.971 ○	0.990 •	0.989 •	0.995 •	0.991 •	0.993 •	0.988 •	0.992 •	0.996 •	0.995 •	0.996 •	0.957 •	0.951 ○
Comparing algorithm	*Intersection similarity* ↑												
	SJ	spoem	spo5	dtt	cold	heat	spo	diau	elu	cdc	alpha	3DFE	Mov
LP	0.837	0.837	0.886	0.786	0.794	0.805	0.819	0.788	0.782	0.779	0.774	0.810	0.778
ML	0.661	0.597	0.727	0.546	0.565	0.564	0.580	0.593	0.544	0.538	0.537	0.587	0.779
GLLE	0.872	0.912	0.901	0.939	0.924	0.929	0.909	0.906	0.936	0.937	0.938	0.850	0.831
LESC	0.905 ○	0.913	0.908	0.949	0.935	0.934	0.912	0.933	0.949	0.950	0.953	0.855	0.833
PLEML	0.858	0.911	0.908	0.957 ○	0.974 •	0.939	0.913	0.933	0.958	0.957	0.962 •	0.859	0.768
LEVI-MLP	0.899	0.937 ○	0.933 ○	0.958 •	0.940 ○	0.952 •	0.940 ○	0.942 ○	0.959 ○	0.958 ○	0.960	0.882 ○	0.850 ○
LEVI-GCN	0.908 •	0.939 •	0.936 •	0.958 •	0.940 ○	0.951 ○	0.941 •	0.946 •	0.960 •	0.959 •	0.961 ○	0.884 •	0.851 •



(a) *Chebyshev distance* (b) *Intersection similarity* (c) *Chebyshev distance* (d) *Intersection similarity*

Fig. 4: Parameter sensitivity analysis for LEVI-MLP and LEVI-GCN on `Yeast-spoem`, `SBD_3DFE` and `Movie`. **(a) and (b)**: Performance of LEVI-MLP changes in terms of two evaluation metrics as the parameter λ increases from 0.4 to 1.6. **(c) and (d)**: Performance of LEVI-GCN changes in terms of two evaluation metrics as the parameter λ increases from 0.4 to 1.6.

4.1.4 Recovery Performance

The description degrees of the three labels in the artificial dataset are treated as the three color channels to show the results of LE approaches visually. Thus the color of each point could represent the label distribution visually and then the label distributions recovered by the LE approaches would be compared with the ground-truth label distributions with the color patterns. The results visually enhanced by adopting a decorrelation stretch process for easier comparison are shown in Fig. 3. It can be seen that LEVI-MLP and LEVI-GCN recover almost identical color patterns against the ground-truth label distributions.

Table 2 tabulates the results of each LE approach on all real-world datasets for quantitative analysis, where • and ○ denote the best and second best performance among all the approaches respectively. There is no record of standard deviation since each LE approach only runs once. LEVI-GCN ranks 1st in 84.62% cases and ranks 2nd in 13.46% cases while LEVI-MLP ranks 1st in 28.85% cases and ranks 2nd in 65.38% cases on all evaluation metrics. We can find that LEVI-MLP and LEVI-GCN achieve superior performance than other LE approaches in terms of all the six evaluation metrics.

4.1.5 Sensitivity Analysis

In this subsection, the performance sensitivity of LEVI-GCN and LEVI-MLP for label enhancement w.r.t. the parameter λ will be further analyzed. Fig. 4 shows the performance of LEVI-MLP and LEVI-GCN under different parameter configurations on three datasets `Yeast-spoem`, `SBU_3DFE`, and `Movie`. It is obvious that the performance of LEVI-GCN and LEVI-MLP is relatively stable across a broad range of parameter λ. This property is quite desirable as one can

TABLE 3: Statistics of the 14 datasets utilized in multi-label prediction experiment

No.	Dataset	#Examples	#Features	#Labels
1	cal500	502	68	174
2	emotion	593	72	6
3	medical	978	1,449	45
4	llog	1,460	1,004	75
5	enron	1,702	1,001	53
6	msra	1,868	898	19
7	image	2,000	294	5
8	scene	2,407	294	6
9	yeast	2,417	103	14
10	slashdot	3,782	1,079	22
11	corel5k	5,000	499	374
12	rcv1subset1	6,000	944	101
13	rcv1subset2	6,000	944	101
14	bibtex	7,395	1,836	159

make use of LEVI to achieve robust label enhancement performance.

4.2 Multi-Label Prediction

In this experiment, the effective performance of LEVI for multi-label learning can be validated. As mentioned in Section 3.3, the multi-label predictive models are induced by the label distributions recovered by LEVI-MLP and LEVI-GCN, which enable the comparison with the predictive performance of the state-of-the-art MLL approaches.

4.2.1 Datasets

There are fourteen multi-label learning datasets[2] utilized in the experiments, which cover a broad range of cases with diversified multi-label properties and thus serve as a solid basis for thorough comparative studies. In addition, these datasets cover a broad range of scenarios, including text (`medical`, `llog`, `enron`, `slashdot`, `rcv1subset1`, `rcv1subset2`, and `bibtex`), audio (`cal500` and `emotions`), image (`image`, `msra`, `scene`, and `corel5k`), and biology (`yeast`). The basic statistics about these datasets are given in Table 3.

4.2.2 Evaluation Metrics

Five popular multi-label metrics including *Ranking loss*, *Hamming loss*, *One-error*, *Coverage*, and *Average precision* [57] are employed for performance evaluation. Let $\mathcal{S} = \{(\mathbf{x}_i, Y_i) \mid 1 \leq i \leq n\}$ be a multi-label test set, Y_i and Z_i be the sets of true and predicted labels for an instance, and $\tau_{\boldsymbol{x}}$ be the rank function which maps the output real value of the classifier to the position of the label in the ranking.

- *Hamming loss*: $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{c}|Z_i \Delta Y_i|$ where Δ stands for the symmetric difference of two sets. *Hamming loss* evaluates how many times, on average, an example-label pair is misclassified. This metric takes into account both prediction errors and omission errors.
- *One-error*: $\frac{1}{n}\sum_{i=1}^{n}[\![\arg\min_{y\in\mathcal{Y}}\tau_i(y) \notin Y_i]\!]$, where $[\![\pi]\!]$ returns 1 if π is true and 0 otherwise. *One-error* measures the fraction of examples whose top-ranked predicted label is not in the ground-truth relevant label set.
- *Coverage*: $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{c}\max_{y\in Y_i}\tau_i(y) - 1$. *Coverage* evaluates how many steps are needed on average to move down the ranked label list of an example so as to cover all its relevant class labels.
- *Ranking loss*: $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i||\bar{Y}_i|}|E|$, where $E = \{(y, y') \mid \tau_i(y) > \tau_i(y'), (y, y') \in Y_i \times \bar{Y}_i\}$. *Ranking loss* evaluates the average fraction of misordered label pairs, i.e., a relevant label of an example is ranked lower than its irrelevant one.
- *Average precision*: $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{|Y_i|}\sum_{y\in Y_i}\frac{|\{y'\in Y_i|\tau_i(y')\leq\tau_i(y)\}|}{\tau_i(y)}$. *Average precision* evaluates the average fraction of labels ranked above a particular label, $y \in Y$, which actually are in Y.

Note that the values of all the five metrics vary between $[0, 1]$. Besides, for *average precision*, the *larger* the better; While for the other four metrics, the *smaller* the better. The metrics could be adopted as well indicators for comprehensive studies since the five metrics evaluate the performance of learned models in different aspects.

4.2.3 Comparing Algorithms

In this paper, LEVI-MLP and LEVI-GCN are compared against seven well-established multi-label learning algorithms which have been widely employed for comparative studies in multi-label learning.

- BR [1] disassembles the multi-label learning problem into c independent binary classification problems, where each of them refer to one class.
- CLR [11] transform the multi-label learning problem into the label ranking problem, where each classifier learns to generate the ranking among the labels and bipartition.
- ECC [34] transform the multi-label learning problem into a series of binary classification problems, where the outputs of binary classifiers are used as extra features to build subsequence [suggested configuration: ensemble size $m = 30$].
- RAKEL [45] transforms the multi-label learning problem into an ensemble of classification problems, where each classifiers is induced by adopting the label powerset techniques on a random k-label set [suggested configuration: ensemble size $m = 2c$, $k = 3$].
- GLOCLA [62] exploits global and local label correlations simultaneously via learning a latent label representation and optimizing label manifolds [suggested configuration: parameter $\lambda = 0.1$].
- RELIAB-LP [55] generates the implicit relative labeling-importance via global label propagation procedure to train a multi-label predictive model with multi-label empirical loss regularization.
- RELIAB-KNN [55] generates the implicit relative labeling-importance via local k-nearest neighbor reconstruction to train multi-label a predictive model with multi-label empirical loss regularization.

We employ three-layer MLP and two-layer GCN as the encoding models of LEVI-MLP and LEVI-GCN, respectively, and three-layer MLP as the decoding models of LEVI-MLP and LEVI-GCN. The numbers of hidden-layer nodes in MLP and GCN are set to 500. The hyper-parameter λ is set to 1.

2. mulan.sourceforge.net/datasets.html

TABLE 4: Predictive performance of each approach (mean±std) measured by *Ranking loss* ↓. The best and second best performance among all the approaches are denoted by • and ○ respectively.

Datasets	LEVI-GCN	LEVI-MLP	RELIAB-LP	RELIAB-KNN	GLOCLA	BR	CLR	ECC	RAKEL
cal500	0.177±0.002 •	0.177±0.002 •	0.181±0.003	0.185±0.003	0.180±0.002 ○	0.258±0.003	0.239±0.026	0.205±0.004	0.444±0.005
emotions	0.183±0.009 •	0.192±0.008	0.185±0.015 ○	0.246±0.017	0.284±0.021	0.233±0.016	0.222±0.014	0.227±0.017	0.254±0.020
medical	0.023±0.004 •	0.024±0.004 ○	0.033±0.006	0.027±0.005	0.049±0.008	0.091±0.005	0.123±0.026	0.032±0.007	0.095±0.033
llog	0.138±0.006	0.154±0.005	0.116±0.005 •	0.120±0.005 ○	0.219±0.008	0.328±0.007	0.190±0.015	0.154±0.009	0.412±0.010
enron	0.079±0.002 •	0.080±0.003 ○	0.093±0.003	0.092±0.004	0.157±0.004	0.312±0.009	0.089±0.002	0.120±0.004	0.241±0.005
msra	0.125±0.010 •	0.126±0.010 ○	0.142±0.012	0.141±0.013	0.135±0.011	0.368±0.021	0.288±0.018	0.332±0.047	0.223±0.075
image	0.141±0.005 •	0.142±0.006 ○	0.180±0.007	0.180±0.006	0.180±0.008	0.314±0.014	0.294±0.009	0.276±0.005	0.311±0.010
scene	0.062±0.003 •	0.062±0.004 ○	0.089±0.005	0.096±0.003	0.098±0.004	0.229±0.010	0.127±0.003	0.151±0.005	0.205±0.008
yeast	0.167±0.002 •	0.169±0.002 ○	0.367±0.006	0.358±0.005	0.356±0.003	0.190±0.004	0.198±0.003	0.190±0.003	0.245±0.004
slashdot	0.094±0.003 •	0.098±0.002 ○	0.137±0.003	0.131±0.002	0.179±0.003	0.240±0.008	0.260±0.007	0.123±0.004	0.190±0.005
corel5k	0.110±0.002 •	0.118±0.002	0.115±0.002	0.110±0.002 •	0.180±0.002	0.416±0.003	0.114±0.002 ○	0.292±0.003	0.627±0.004
rcv1subset1	0.036±0.001 •	0.040±0.001	0.045±0.001	0.038±0.001 ○	0.099±0.003	0.279±0.004	0.040±0.001	0.079±0.002	0.243±0.004
rcv1subset2	0.037±0.001 •	0.038±0.001 ○	0.042±0.001	0.039±0.001	0.111±0.003	0.251±0.004	0.042±0.001	0.096±0.004	0.216±0.004
bibtex	0.063±0.001 •	0.065±0.001 ○	0.208±0.004	0.207±0.008	0.129±0.004	0.303±0.004	0.065±0.002	0.192±0.003	0.286±0.003

TABLE 5: Predictive performance of each approach (mean±std) measured by *Hamming loss* ↓. The best and second best performance among all the approaches are denoted by • and ○ respectively.

Datasets	LEVI-GCN	LEVI-MLP	RELIAB-LP	RELIAB-KNN	GLOCLA	BR	CLR	ECC	RAKEL
cal500	0.139±0.002	0.137±0.002 •	0.186±0.002	0.191±0.003	0.149±0.002	0.214±0.004	0.165±0.005	0.146±0.002	0.138±0.002 ○
emotions	0.221±0.014 •	0.224±0.008 ○	0.332±0.039	0.317±0.021	0.311±0.005	0.265±0.013	0.270±0.011	0.254±0.013	0.269±0.011
medical	0.010±0.001 •	0.012±0.001	0.015±0.001	0.016±0.001	0.028±0.000	0.022±0.003	0.024±0.002	0.013±0.001	0.010±0.003 ○
llog	0.015±0.000 •	0.015±0.000 •	0.015±0.000 •	0.017±0.001	0.018±0.000	0.052±0.003	0.019±0.002	0.016±0.000 ○	0.017±0.001
enron	0.047±0.001 •	0.047±0.001 •	0.064±0.003	0.075±0.003	0.065±0.001	0.105±0.003	0.072±0.002	0.064±0.001	0.058±0.001 ○
msra	0.180±0.008 •	0.182±0.009 ○	0.231±0.015	0.218±0.014	0.670±0.005	0.404±0.037	0.342±0.033	0.353±0.037	0.237±0.079
image	0.154±0.003 •	0.157±0.003 ○	0.245±0.018	0.214±0.005	0.247±0.002	0.287±0.008	0.305±0.005	0.244±0.005	0.286±0.007
scene	0.078±0.003 •	0.080±0.002 ○	0.184±0.008	0.175±0.007	0.178±0.000	0.184±0.005	0.181±0.004	0.133±0.002	0.171±0.005
yeast	0.194±0.003 •	0.195±0.003 ○	0.433±0.005	0.433±0.004	0.302±0.002	0.219±0.003	0.222±0.002	0.216±0.002	0.202±0.003
slashdot	0.038±0.001 •	0.039±0.001 ○	0.067±0.001	0.065±0.002	0.053±0.000	0.130±0.003	0.058±0.001	0.049±0.001	0.048±0.001
corel5k	0.009±0.000 •	0.009±0.000 •	0.010±0.000 ○	0.010±0.000 ○	0.009±0.000 •	0.027±0.000	0.011±0.001	0.015±0.001	0.012±0.001
rcv1subset1	0.026±0.000 •	0.026±0.000 •	0.027±0.001 ○	0.034±0.002	0.028±0.000	0.031±0.001	0.029±0.001	0.030±0.001	0.031±0.001
rcv1subset2	0.023±0.000 •	0.023±0.000 •	0.027±0.001	0.030±0.001	0.026±0.000	0.028±0.001	0.025±0.001	0.024±0.001 ○	0.027±0.001
bibtex	0.013±0.000 •	0.013±0.000 •	0.015±0.000	0.015±0.000	0.015±0.000	0.015±0.001	0.014±0.001 ○	0.017±0.001	0.015±0.001

TABLE 6: Predictive performance of each approach (mean±std) measured by *Average precision* ↑. The best and second best performance among all the approaches are denoted by • and ○ respectively.

Datasets	LEVI-GCN	LEVI-MLP	RELIAB-LP	RELIAB-KNN	GLOCLA	BR	CLR	ECC	RAKEL
cal500	0.512±0.004 •	0.511±0.004 ○	0.495±0.004	0.493±0.006	0.503±0.005	0.300±0.005	0.395±0.042	0.463±0.006	0.353±0.006
emotions	0.781±0.010 •	0.773±0.008 ○	0.772±0.018	0.720±0.011	0.674±0.021	0.730±0.015	0.742±0.016	0.740±0.021	0.717±0.023
medical	0.893±0.011 •	0.879±0.014 ○	0.838±0.016	0.858±0.011	0.847±0.014	0.762±0.022	0.400±0.062	0.860±0.015	0.700±0.234
llog	0.409±0.011 •	0.367±0.013	0.399±0.011 ○	0.382±0.010	0.366±0.008	0.215±0.009	0.194±0.018	0.342±0.009	0.197±0.013
enron	0.698±0.009 •	0.697±0.008 ○	0.656±0.007	0.661±0.011	0.609±0.009	0.381±0.009	0.610±0.008	0.559±0.008	0.539±0.006
msra	0.827±0.013 •	0.826±0.013 ○	0.805±0.015	0.804±0.016	0.814±0.014	0.540±0.015	0.624±0.022	0.567±0.048	0.601±0.200
image	0.828±0.006 •	0.824±0.005 ○	0.779±0.007	0.782±0.006	0.781±0.009	0.649±0.012	0.666±0.008	0.685±0.008	0.661±0.010
scene	0.888±0.004 •	0.887±0.005 ○	0.841±0.006	0.832±0.003	0.835±0.005	0.692±0.010	0.778±0.004	0.766±0.005	0.713±0.008
yeast	0.766±0.005 •	0.765±0.005 ○	0.601±0.005	0.607±0.005	0.599±0.004	0.734±0.004	0.730±0.003	0.741±0.004	0.720±0.005
slashdot	0.711±0.006 •	0.710±0.005 ○	0.565±0.007	0.596±0.007	0.602±0.006	0.427±0.014	0.250±0.007	0.628±0.009	0.617±0.004
corel5k	0.301±0.003 •	0.297±0.003 ○	0.258±0.003	0.275±0.003	0.269±0.002	0.123±0.003	0.274±0.002	0.264±0.003	0.122±0.004
rcv1subset1	0.632±0.004 •	0.625±0.003	0.592±0.007	0.613±0.005	0.533±0.007	0.383±0.007	0.628±0.003 ○	0.606±0.004	0.436±0.006
rcv1subset2	0.649±0.005 •	0.642±0.004 ○	0.620±0.005	0.640±0.004	0.534±0.007	0.434±0.005	0.641±0.003	0.616±0.005	0.487±0.005
bibtex	0.577±0.003 ○	0.583±0.004 •	0.334±0.013	0.343±0.015	0.430±0.003	0.363±0.004	0.564±0.004	0.515±0.004	0.399±0.004

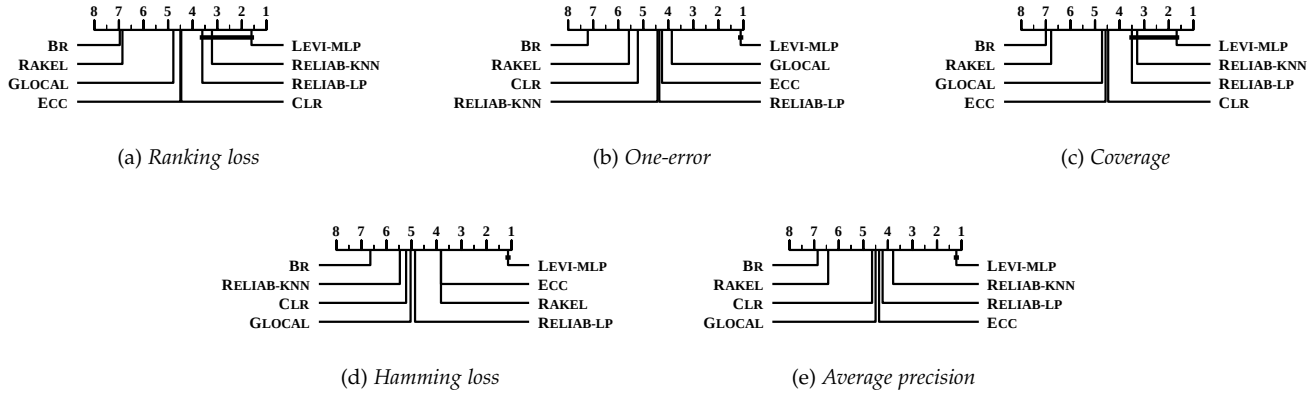
We use Adam as the optimizer and the learning rate and the weight decay are set to 1e-3 and 1e-5, respectively. For LEVI-MLP and LEVI-GCN, the parameters β and γ are set to 1 and 0.01, respectively. The kernel function of each approach is Gaussian kernel.

4.2.4 Predictive Performance

Table 4 to 6 tabulate the results of all the algorithms (LEVI-MLP, LEVI-GCN, BR, CLR, ECC, RAKEL, GLOCLA, RELIAB-LP and RELIAB-KNN) on the fourteen multi-label learning datasets evaluated by *Ranking loss*, *Hamming loss* and *Average precision*, where • and ○ denote the best and second best performance among all the approaches respectively. The results on other evaluation measures are similar. For each evaluation metric, ↓ indicates the smaller the better while ↑ indicates the larger the better. Ten-fold cross-validation is adopted for all approaches.

In addition, the *Friedman test* [7] is adopted to analyze the relative performance of these methods. At 0.05 significance level, the Friedman statistics F_F ($F_F > 16$ on all evaluation metrics) is greater than the critical value 2.70 (#algorithms $n = 8$; #datasets $N = 14$). Therefore, the null hypothesis of indistinguishable performance among comparing approaches is rejected on all of the evaluation metrics across the 14 benchmark cases.

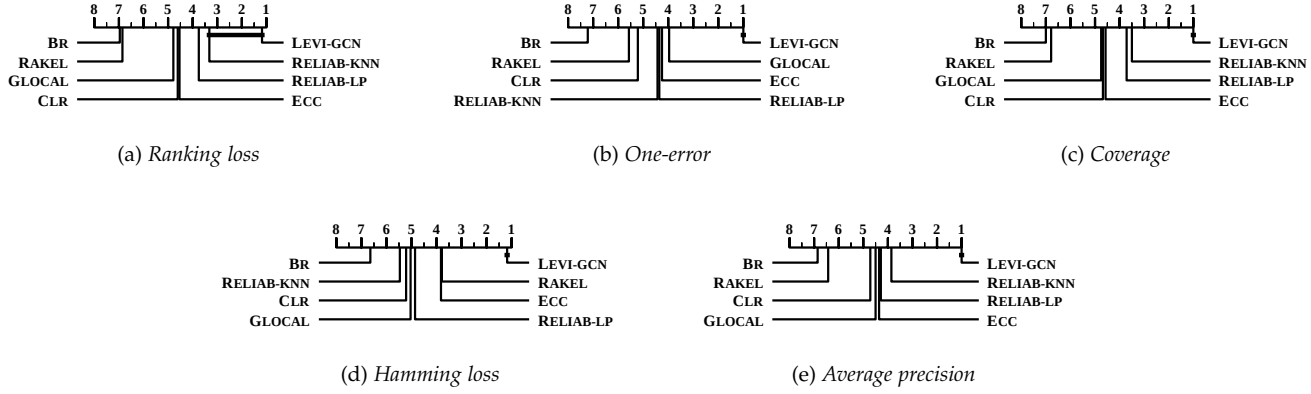
Bonferroni-Dunn test [7] is utilized as the post-hoc test to show whether the proposed approaches have a significantly different performance against comparing approaches. Here, LEVI-GCN and LEVI-MLP are regarded as the control approaches, and the *critical difference* (CD) calibrates the



(a) *Ranking loss* (b) *One-error* (c) *Coverage*

(d) *Hamming loss* (e) *Average precision*

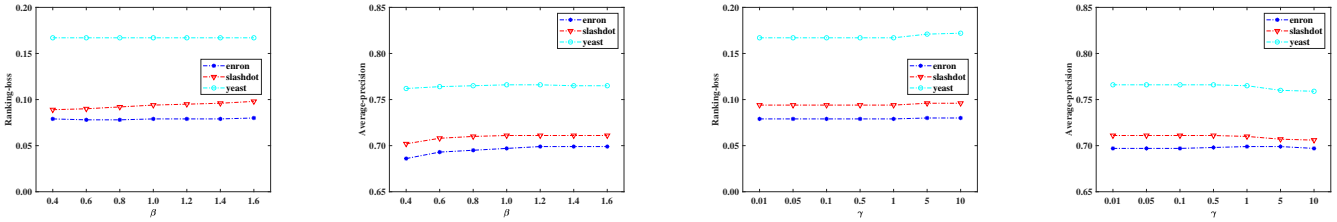
Fig. 5: Comparison of LEVI-MLP against other comparing approaches with the *Bonferroni-Dunn test*. The approaches not connected with LEVI-MLP are considered to be significantly different from LEVI-MLP (CD=2.4905 at 0.05 significance level).



(a) *Ranking loss* (b) *One-error* (c) *Coverage*

(d) *Hamming loss* (e) *Average precision*

Fig. 6: Comparison of LEVI-GCN against other comparing approaches with the *Bonferroni-Dunn test*. The approaches not connected with LEVI-GCN are considered to be significantly different from LEVI-GCN (CD=2.4905 at 0.05 significance level).



(a) *Ranking loss* (b) *Average precision* (c) *Ranking loss* (d) *Average precision*

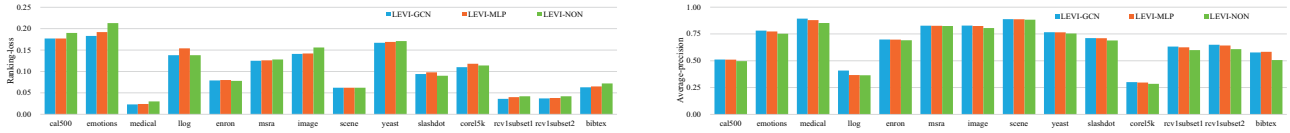
Fig. 7: Parameter sensitivity analysis for LEVI on `enron`, `slashdot` and `yeast`. **(a) and (b)**: Performance changes in terms of *Ranking loss* and *Average precision* as the parameter β increases from 0.4 to 1.6 ($\gamma = 0.01$). **(c) and (d)**: Performance changes in terms of *Ranking loss* and *Average precision* as the parameter γ increases from 0.01 to 10 ($\beta = 1$).

difference of the average rank over all datasets between the control approach and one comparing approach. Then, the performance between the control approach and one comparing approach is considered to be significantly different if difference of the average rank is greater than the CD (CD = 2.4905 with comparing approaches $n = 8$, and benchmark datasets $N = 14$),

Fig. 5 and 6 illustrate the CD diagrams [7] in terms of five evaluation metrics. Here, the average ranks of the approaches are marked along the axis. If the average rank difference between the control approach and one comparing approach is within the CD, we use a thick line to connect them. Otherwise, the control approach is considered to be significantly different from the comparing approach.

Based on the experimental results of the comparative studies, we could make the following observations:

- LEVI-GCN ranks 1st in 91.43% cases and ranks 2nd in 0.06% cases while LEVI-MLP ranks 1st in 18.57% cases and ranks 2nd in 61.43% cases on all evaluation metrics.
- As shown in Fig. 5 and 6, both LEVI-GCN and LEVI-MLP achieve optimal (lowest) average rank on all the evaluation metrics. Specifically, LEVI-GCN achieve superior performance against BR, ECC, CLR, RAKEL, and RELIAB-LP on all evaluation metrics and LEVI-MLP



(a) *Ranking loss* ↓ (b) *Average precision* ↑

Fig. 8: Performance comparison among LEVI-GCN, LEVI-MLP and LEVI-NON in terms of *Ranking loss* and *Average precision*.

TABLE 7: Wilcoxon signed-ranks test for LEVI against its variant LEVI-NON in terms of each evaluation metric (at 0.05 significance level) and p-values are shown in the brackets.

Evaluation metric	LEVI-MLP	LEVI-GCN
	against LEVI-NON	
Ranking loss	**tie** [p=2.47e-1]	**win** [p=7.60e-3]
One-error	**win** [p=1.50e-3]	**win** [p=9.79e-4]
Coverage	**tie** [p=5.30e-1]	**win** [p=1.69e-2]
Hamming loss	**win** [p=3.30e-3]	**win** [p=2.20e-3]
Average precision	**win** [p=9.76e-4]	**win** [p=9.79e-4]

achieve superior performance against BR, ECC, CLR, and RAKEL on all evaluation metrics.

- LEVI-GCN achieves superior performance against RELIAB-KNN on *One-error*, *Hamming loss* and *Average precision*, and is comparable to RELIAB-KNN on *Ranking loss*. LEVI-MLP achieves superior performance against RELIAB-LP and RELIAB-KNN on *One-error*, *Hamming loss* and *Average precision*, and is comparable to RELIAB-LP and RELIAB-KNN on *Ranking loss* and *Coverage*. Note that both RELIAB-LP and RELIAB-KNN consider the implicit ranking information, i.e, relative labeling-importance of each example, which could improve the performance on the rank based evaluation metrics *Ranking loss* and *Coverage*.

4.2.5 Sensitivity Analysis

In this subsection, the performance sensitivity of LEVI-GCN and LEVI-MLP for MLL w.r.t. the parameters β and γ will be further analyzed. Fig. 7 shows the performance of LEVI-GCN under different parameter configurations on three datasets `enron`, `slashdot` and `yeast` while similar observations also hold on other datasets. As shown in Fig. 7, it is obvious that the performance of our approach is relatively stable across a broad range of each parameter. The parameter configuration for LEVI in Subsection 5.2.3 naturally follows from these observations.

4.2.6 Usefulness of Recovered Label Distribution

To illustrate the usefulness of the label distributions to our advantage, a vanilla variant of LEVI for MLL (termed as LEVI-NON) is employed here. LEVI-NON ablates the label enhancement stage and maintains the same procedure in the other stages for MLL. Following the same evaluation protocol of Subsection 5.2.2, the performance of LEVI-NON is investigated as well.

Fig. 8 reports the experimental results on *Ranking loss* and *Average precision* while similar observations also hold on other metrics. To show whether LEVI-MLP and LEVI-GCN perform significantly better than their ablation version, the Wilcoxon signed-ranks test [7] is employed. Table 7 summarizes the statistical test results at 0.05 significance level, where the p-values for the corresponding tests are also shown.

As shown in Table 7, LEVI-MLP achieves superior or at least comparable performance to LEVI-NON across all evaluation metrics. In addition, LEVI-GCN achieves superior performance to LEVI-NON across all evaluation metrics. These results clearly validate the usefulness of recovered label distributions for improving predictive performance.

5 CONCLUSION

Label enhancement is the process of recovering the label distributions from the training examples with logical labels, which can help describe the supervised information in a more fine-grained way for learning with label ambiguity. In this paper, we propose the theoretical explanation of the label enhancement process and two LE approaches. In addition, the multi-label predictive model is induced via leveraging the recovered label distributions. Comprehensive experimental studies validate the performance superiority of proposed methods against state-of-the-art comparing algorithms as well as the usefulness of the recovered label distributions.

In the future, other than variational inference, it is interesting to explore other ways for latent label distribution recovery. It is also interesting to further employ label enhancement with auxiliary information to deal with other learning problems, such as learning with noisy labels, partial label learning, zero-shot learning, etc.

REFERENCES

[1] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[2] C. Brinker, E. L. Mencía, and J. Fürnkranz, "Graded multilabel classification by pairwise comparisons," in *2014 IEEE International Conference on Data Mining*, Shenzhen, China, 2014, pp. 731–736.

[3] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for weakly-supervised multi-label image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 1, pp. 121–135, 2014.

[4] S. Chen, J. Wang, Y. Chen, Z. Shi, X. Geng, and Y. Rui, "Label distribution learning on auxiliary label space graphs for facial expression recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 2020, pp. 13 984–13 993.

[5] W. Chung, J. Kim, H. Lee, and E. Kim, "General dimensional multiple-output support vector regressions and their multiple kernel learning," *IEEE transactions on cybernetics*, vol. 45, no. 11, pp. 2572–2584, 2014.

[6] K. Dahiya, D. Saini, A. Mittal, A. Shaw, K. Dave, A. Soni, H. Jain, S. Agarwal, and M. Varma, "Deepxml: A deep extreme multi-label learning framework applied to short text documents," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, online, 2021, pp. 31–39.

[7] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[8] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[9] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14 863–14 868, 1998.

[10] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in Neural Information Processing Systems 14 (NIPS 2002)*, Vancouver, British Columbia, Canada, 2002, pp. 681–687.

[11] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.

[12] B.-B. Gao, H.-Y. Zhou, J. Wu, and X. Geng, "Age estimation using expectation of label distribution learning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 712–718.

[13] Y. Gao, Y. Zhang, and X. Geng, "Label enhancement for label distribution learning via prior knowledge," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Yokohama, Japan, 2020, pp. 3223–3229.

[14] X. Geng, "Label distribution learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1734–1748, 2016.

[15] X. Geng and P. Hou, "Pre-release prediction of crowd opinion on movies by label distribution learning," in *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[16] X. Geng and L. Luo, "Multilabel ranking with inconsistent rankers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 3742–3747.

[17] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 644–658, 2014.

[18] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, pp. 1–38, 2015.

[19] P. Hou, X. Geng, and M.-L. Zhang, "Multi-label manifold learning," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ, 2016, pp. 1680–1686.

[20] X. Jia, Y. Lu, and F. Zhang, "Label enhancement by maintaining positive and negative label relation," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[21] E.-S. Kim, K.-W. On, J. Kim, Y.-J. Heo, S.-H. Choi, H.-D. Lee, and B.-T. Zhang, "Temporal attention mechanism with conditional inference for large-scale multi-label video classification," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, Munich, Germany, 2018, pp. 1–10.

[22] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, Banff, AB, Canada, 2014.

[23] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[24] ——, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[25] X. Liu, X. Nie, H. Sun, C. Cui, and Y. Yin, "Modality-specific structure preserving hashing for cross-modal retrieval," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 1678–1682.

[26] X. Liu, J. Zhu, Q. Zheng, Z. Li, R. Liu, and J. Wang, "Bidirectional loss function for label enhancement and distribution learning," *Knowledge-Based Systems*, vol. 213, p. 106690, 2020.

[27] H.-Y. Lo, J.-C. Wang, H.-M. Wang, and S.-D. Lin, "Cost-sensitive multi-label learning for audio tag annotation and retrieval," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 518–529, 2011.

[28] X. Long, W. Qian, Y. Wang, and W. Shu, "Cost-sensitive feature selection on multi-label data via neighborhood granularity and label enhancement," *Applied Intelligence*, pp. 1–23, 2020.

[29] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Proceedings Third IEEE international conference on automatic face and gesture recognition*. IEEE, 1998, pp. 200–205.

[30] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, United Kingdom, 2018, pp. 2041–2050.

[31] F. Pérez-Cruz, G. Camps-Valls, E. Soria-Olivas, J. J. Pérez-Ruixo, A. R. Figueiras-Vidal, and A. Artés-Rodríguez, "Multi-dimensional function approximation and regression estimation," in *Proceedings of the International conference on artificial neural networks*, Madrid, Spain, 2002, pp. 757–762.

[32] F. Pérez-Cruz, A. Navia-Vázquez, P. L. Alarcón-Diana, and A. Artés-Rodríguez, "An irwls procedure for svr," in *Proceedings of the 10th European Signal Processing Conference*, Tampere, Finland, 2000, pp. 1–4.

[33] W. Qian, J. Huang, Y. Wang, and W. Shu, "Mutual information-based label distribution feature selection for multi-label learning," *Knowledge-Based Systems*, vol. 195, p. 105684, 2020.

[34] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, p. 333, 2011.

[35] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[36] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers, "Statistical topic models for multi-label document classification," *Machine Learning*, vol. 88, no. 1-2, pp. 157–208, 2012.

[37] R. Shao, N. Xu, and X. Geng, "Multi-label learning with label enhancement," in *2018 IEEE International Conference on Data Mining (ICDM)*, Singapore, 2018, pp. 437–446.

[38] W. Shen, K. Zhao, Y. Guo, and A. Yuille, "Label distribution learning forests," in *Proceedings of the 30th International Conference on Neu-ral Information Processing Systems*, Long Beach, California, 2017, pp. 834–843.

[39] A. J. Smola, *Learning with kernels*. Ph.D. Thesis, GMD, Birlinghoven, German, 1999.

[40] K. Su and X. Geng, "Soft facial landmark detection by label distribution learning," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, 2019, p. in press.

[41] C. Tan, S. Chen, G. Ji, and X. Geng, "A novel probabilistic label enhancement algorithm for multi-label distribution learning," *IEEE Transactions on Knowledge and Data Engineering*, p. in press, 2021.

[42] C. Tan, G. Ji, R. Liu, and Y. Cao, "Ltsa-le: A local tangent space alignment label enhancement algorithm," *Tsinghua Science and Technology*, 2019.

[43] H. Tang, J. Zhu, Q. Zheng, J. Wang, S. Pang, and Z. Li, "Label enhancement with sample correlations via low-rank representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, New York, 2020, pp. 5932–5939.

[44] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2006.

[45] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.

[46] D. Wang and S. Zhang, "Unsupervised person re-identification via multi-label classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 2020, pp. 10 981–10 990.

[47] J. Wang, Y. Zhao, X. Wu, and X.-S. Hua, "A transductive multi-label learning approach for video concept detection," *Pattern Recognition*, vol. 44, no. 10, pp. 2274–2286, 2011.

[48] M. Xu, Y.-F. Li, and Z.-H. Zhou, "Robust multi-label learning with pro loss," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1610–1624, 2019.

[49] N. Xu, Y.-P. Liu, and X. Geng, "Label enhancement for label distribution learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1632 – 1643, 2021.

[50] N. Xu, Y.-P. Liu, Y. Zhang, and X. Geng, "Progressive enhancement of label distributions for partial multilabel learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[51] N. Xu, J. Shu, Y.-P. Liu, and X. Geng, "Variational label enhancement," in *Proceedings of the International Conference on Machine Learning*, Vienna, Austria, 2020, pp. 10 597–10 606.

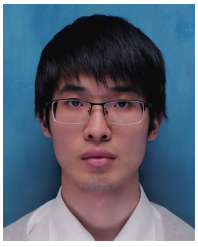
[52] N. Xu, A. Tao, and X. Geng, "Label enhancement for label distribution learning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 2926–2932.

[53] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3d facial expression database for facial behavior research," in *7th international conference on automatic face and gesture recognition (FGR06)*. IEEE, 2006, pp. 211–216.

[54] F. Zhang, X. Jia, and W. Li, "Tensor based multi-view label enhancement for multi-label learning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Yokohama, Japan, 2020, pp. 2369–2375.
[55] M.-L. Zhang, Q.-W. Zhang, J.-P. Fang, Y.-K. Li, and X. Geng, "Leveraging implicit relative labeling-importance information for effective multi-label learning," *IEEE Transactions on Knowledge and Data Engineering*, 2019.
[56] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.
[57] ——, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.
[58] Q.-W. Zhang, Y. Zhong, and M.-L. Zhang, "Feature-induced labeling information enrichment for multi-label learning." in *Proceedings of the AAAI Conference on Artificial Intelligence*, New Orleans, LA, 2018, pp. 4446–4453.
[59] Q. Zheng, J. Zhu, H. Tang, X. Liu, Z. Li, and H. Lu, "Generalized label enhancement with sample correlations," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
[60] D. Zhou, Y. Zhou, X. Zhang, Q. Zhao, and X. Geng, "Emotion distribution learning from texts," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Austin, TX, 2016, pp. 638–647.
[61] W. Zhu, X. Jia, and W. Li, "Privileged label enhancement with multi-label learning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Yokohama, Japan, 2020, pp. 2376–2382.
[62] Y. Zhu, J. T. Kwok, and Z.-H. Zhou, "Multi-label learning with global and local label correlation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1081–1094, 2017.

Ning Xu received the BSc and MSc degrees from University of Science and Technology of China and Chinese Academy of Sciences China, respectively, and the PhD degree from Southeast University, China. He is now an assistant professor in the School of Computer Science and Engineering at Southeast University, China. His research interests mainly include machine learning and data mining, especially on multi-label learning and weakly-supervised learning.

Shu Jun received the B.E. degree from Xi'an Jiaotong University, Xi'an, China, in 2016, where he is currently pursuing the Ph.D degree, under the tuition of Prof. Deyu Meng and Prof. Zongben Xu. His current research interests include machine learning and computer vision, especially on meta learning, robust deep learning and AutoML.



Renyi Zheng received the B.Sc. degree in computer science from Soochow University, China, in 2018. He is currently a master student in the School of Computer Science and Engineering at Southeast University, China. His research interests include machine learning and computer vision.

Xin Geng is currently a professor and the dean of School of Computer Science and Engineering at Southeast University, China. He received the B.Sc. (2001) and M.Sc. (2004) degrees in computer science from Nanjing University, China, and the Ph.D. (2008) degree in computer science from Deakin University, Australia. His research interests include machine learning, pattern recognition, and computer vision. He has published over 90 refereed papers in these areas, including those published in prestigious journals and top international conferences. He has been an Associate Editor of IEEE T-MM, Electronics, FCS and MFC, a Steering Committee Member of PRICAI, a Program Committee Chair for conferences such as PRICAI'18, VALSE'13, etc., an Area Chair for conferences such as IJCAI'21, CVPR'21, ACMMM'18, ICPR'21, WACV'21, and a Senior Program Committee Member for conferences such as AAAI, ECAI, etc. He is a Distinguished Fellow of IETI and a Member of IEEE.

Deyu Meng received the B.Sc., M.Sc., and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2001, 2004, and 2008, respectively. He was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA, from 2012 to 2014. He is currently a Professor with the School of Mathematics and Statistics, Xi'an Jiaotong University, and an Adjunct Professor with the Faculty of Information Technology, Macau University of Science and technology, Taipa, Macau, China. His research interests include model-based deep learning, variational networks, and meta learning.

Min-Ling Zhang received the BSc, MSc, and PhD degrees in computer science from Nanjing University, China, in 2001, 2004 and 2007, respectively. Currently, he is a Professor at the School of Computer Science and Engineering, Southeast University, China. His main research interests include machine learning and data mining. In recent years, Dr. Zhang has served as the General Co-Chairs of ACML'18, Program Co-Chairs of PAKDD'19, CCF-ICAI'19, ACML'17, CCFAI'17, PRICAI'16, Senior PC member or Area Chair of KDD 2021, AAAI 2017-2020, IJCAI 2017-2021, ICDM 2015-2020, etc. He is also on the editorial board of ACM Transactions on Intelligent Systems and Technology, Neural Networks, Science China Information Sciences, Frontiers of Computer Science, etc. Dr. Zhang is the Steering Committee Member of ACML and PAKDD, secretary-general of the CAAI Machine Learning Society, standing committee member of the CCF Artificial Intelligence & Pattern Recognition Society. He is a Distinguished Member of CCF, CAAI, and Senior Member of ACM, IEEE.